INSEL

Block Reference

INSEL 8 :: Block Reference

Updated in 2021 by Eric Duminil.

Please visit us at www.insel.eu

Contents

1 Fundamental Blocks **3**

1.1 Block ABS . . . 5
1.2 Block ACC . . . 6
1.3 Block ACCC . . . 8
1.4 Block ACCP . . . 10
1.5 Block ACOS . . . 12
1.6 Block ACOT . . . 14
1.7 Block AND . . . 16
1.8 Block ANINT . . . 18
1.9 Block ASIN . . . 20
1.10 Block ATAN . . . 22
1.11 Block ATAN2 . . . 24
1.12 Block ATEND . . . 25
1.13 Block ATT . . . 27
1.14 Block AVE . . . 28
1.15 Block AVEC . . . 30
1.16 Block AVEM . . . 32
1.17 Block AVEP . . . 34
1.18 Block AVEW . . . 36
1.19 Block CHARTXY . . . 37
1.20 Block CHS . . . 38
1.21 Block CLOCK . . . 39
1.22 Block CLOCKL . . . 42
1.23 Block CONST . . . 44
1.24 Block COS . . . 45
1.25 Block COSH . . . 47
1.26 Block COT . . . 49
1.27 Block COTH . . . 51
1.28 Block CSCH . . . 53
1.29 Block CUM . . . 55
1.30 Block CUMC . . . 57
1.31 Block CUMP . . . 59

1.32 Block CYCLEF . . . 61
1.33 Block DEG2RAD . . . 63
1.34 Block DELAY . . . 64
1.35 Block DELAYS . . . 66
1.36 Block DIFF . . . 68
1.37 Block DISEXP . . . 69
1.38 Block DISGAU . . . 71
1.39 Block DISRAY . . . 73
1.40 Block DISWEI . . . 75
1.41 Block DIV . . . 77
1.42 Block DLC . . . 79
1.43 Block DO . . . 81
1.44 Block DOW . . . 83
1.45 Block DOY . . . 85
1.46 Block DPM . . . 87
1.47 Block E . . . 89
1.48 Block EQ . . . 90
1.49 Block ERF . . . 92
1.50 Block EXP . . . 94
1.51 Block EXPG . . . 96
1.52 Block EXPRESSION . . . 98
1.53 Block F . . . 99
1.54 Block FDIST . . . 100
1.55 Block FFT . . . 102
1.56 Block FITEXP . . . 104
1.57 Block FITLIN . . . 106
1.58 Block FITLLS . . . 109
1.59 Block FITLN . . . 110
1.60 Block FITPOW . . . 112
1.61 Block FITWEI . . . 114
1.62 Block FRAC . . . 116
1.63 Block GAIN . . . 118
1.64 Block GASDEV . . . 119
1.65 Block GE . . . 121
1.66 Block GREGOR . . . 123
1.67 Block GS . . . 125
1.68 Block GT . . . 126
1.69 Block H . . . 128
1.70 Block HBAR . . . 129
1.71 Block HOY . . . 130
1.72 Block HYSTA . . . 132
1.73 Block HYSTC . . . 134
1.74 Block IF . . . 136

1.75 Block IFELSENAN . . . 138
1.76 Block IFNEG . . . 139
1.77 Block IFPOS . . . 140
1.78 Block INFINITY . . . 141
1.79 Block INT . . . 142
1.80 Block INTGRL . . . 144
1.81 Block INV . . . 146
1.82 Block JULIAN . . . 148
1.83 Block K . . . 150
1.84 Block LE . . . 151
1.85 Block LN . . . 153
1.86 Block LOG . . . 155
1.87 Block LOOP . . . 157
1.88 Block LT . . . 160
1.89 Block MAE . . . 162
1.90 Block MAX . . . 163
1.91 Block MAXX . . . 165
1.92 Block MAXXC . . . 166
1.93 Block MAXXP . . . 168
1.94 Block MAXXXY . . . 170
1.95 Block MAXXXYC . . . 172
1.96 Block MAXXXYP . . . 174
1.97 Block MBE . . . 176
1.98 Block MBOX . . . 177
1.99 Block MBROT . . . 179
1.100 Block MIN . . . 181
1.101 Block MINN . . . 183
1.102 Block MINNC . . . 185
1.103 Block MINNP . . . 187
1.104 Block MINNXY . . . 189
1.105 Block MINNXYC . . . 191
1.106 Block MINNXYP . . . 193
1.107 Block MOD . . . 195
1.108 Block MOY . . . 196
1.109 Block MPLEX . . . 198
1.110 Block MPP . . . 200
1.111 Block MUL . . . 203
1.112 Block NAN . . . 204
1.113 Block NE . . . 205
1.114 Block NOP . . . 207
1.115 Block NOW . . . 208
1.116 Block NULL . . . 210
1.117 Block OFFSET . . . 212

1.118 Block OR . . . 213
1.119 Block PI . . . 215
1.120 Block PID . . . 216
1.121 Block PLOT . . . 218
1.122 Block PLOTP . . . 222
1.123 Block PLOTPM3D . . . 224
1.124 Block PLOTPMC . . . 226
1.125 Block POLYG . . . 228
1.126 Block POLYG2 . . . 230
1.127 Block POLYGS . . . 233
1.128 Block POLYN . . . 235
1.129 Block Q . . . 237
1.130 Block RAD2DEG . . . 238
1.131 Block RAN1 . . . 239
1.132 Block READ . . . 241
1.133 Block READD . . . 243
1.134 Block READN . . . 246
1.135 Block REPORT . . . 248
1.136 Block RMAE . . . 251
1.137 Block RMBE . . . 252
1.138 Block RMSE . . . 253
1.139 Block ROOT . . . 254
1.140 Block RRMSE . . . 256
1.141 Block SCH . . . 257
1.142 Block SCREEN . . . 259
1.143 Block SCREEN1G . . . 261
1.144 Block SIGMA . . . 262
1.145 Block SIN . . . 263
1.146 Block SINH . . . 265
1.147 Block SOY . . . 267
1.148 Block SPEAR . . . 269
1.149 Block SQRT . . . 270
1.150 Block STDEV . . . 272
1.151 Block STDEVC . . . 275
1.152 Block STDEVP . . . 277
1.153 Block STOP . . . 279
1.154 Block SUM . . . 281
1.155 Block SUMP . . . 282
1.156 Block TAN . . . 284
1.157 Block TANH . . . 286
1.158 Block TOL . . . 288
1.159 Block ULC . . . 289
1.160 Block VECTOR . . . 291

1.161 Block WRITE . . . 292
1.162 Block XOR . . . 295

2 Energy Meteorology 297
2.1 Block AM . . . 299
2.2 Block ATM1976 . . . 301
2.3 Block BETAMAX . . . 303
2.4 Block BETAOPT . . . 305
2.5 Block BETGAM . . . 307
2.6 Block COWV . . . 309
2.7 Block DINCLI . . . 311
2.8 Block DISGR . . . 312
2.9 Block E0 . . . 314
2.10 Block ET . . . 316
2.11 Block G2GDD . . . 319
2.12 Block G2GDH . . . 322
2.13 Block G2GDM . . . 325
2.14 Block GBN2GBT . . . 328
2.15 Block GENG . . . 330
2.16 Block GENG2 . . . 333
2.17 Block GENGD . . . 335
2.18 Block GENGD2 . . . 338
2.19 Block GENGH . . . 340
2.20 Block GENGH2 . . . 343
2.21 Block GENGT . . . 345
2.22 Block GENGT2 . . . 348
2.23 Block GENV . . . 350
2.24 Block GH2GT . . . 352
2.25 Block GH2GT2 . . . 355
2.26 Block GMEAN . . . 357
2.27 Block GOH . . . 360
2.28 Block GOH2 . . . 364
2.29 Block GON . . . 365
2.30 Block GTLOSS . . . 367
2.31 Block INANGE . . . 370
2.32 Block INANGH . . . 372
2.33 Block KN . . . 374
2.34 Block MOONAE2 . . . 376
2.35 Block MTM . . . 378
2.36 Block MTM2 . . . 381
2.37 Block MTMLALO . . . 382
2.38 Block MTMLALO2 . . . 384
2.39 Block MTMUP . . . 386

2.40 Block PHIOPT . . . 389
2.41 Block PLANCK . . . 391
2.42 Block SEDES3 . . . 393
2.43 Block SHADF . . . 395
2.44 Block SKYALL . . . 397
2.45 Block SKYC . . . 398
2.46 Block SKYIC . . . 400
2.47 Block SKYIM . . . 402
2.48 Block SKYIO . . . 404
2.49 Block SKYO . . . 406
2.50 Block SUNAE . . . 408
2.51 Block SUNAE2 . . . 411
2.52 Block SUNAEA . . . 413
2.53 Block SUNAEA2 . . . 414
2.54 Block TAUW . . . 415
2.55 Block TDEW . . . 417
2.56 Block TMEAN . . . 419
2.57 Block TSKY . . . 421
2.58 Block TSOIL . . . 423

3 Solar Electricity 427
3.1 Block BALON . . . 429
3.2 Block BCR . . . 432
3.3 Block BDC . . . 435
3.4 Block BTI . . . 437
3.5 Block BTV . . . 441
3.6 Block BUCKET . . . 444
3.7 Block CABLE . . . 446
3.8 Block COMPRS . . . 448
3.9 Block CONTRL . . . 450
3.10 Block CPUMP . . . 452
3.11 Block CUTOFF . . . 454
3.12 Block DIODEI . . . 456
3.13 Block DIODEV . . . 458
3.14 Block ECV . . . 460
3.15 Block ESP . . . 466
3.16 Block ETAIS . . . 468
3.17 Block FCV . . . 469
3.18 Block FERMID . . . 474
3.19 Block HPPOLY . . . 476
3.20 Block IVETAEU . . . 478
3.21 Block IVP . . . 480
3.22 Block IVPFIT . . . 482

3.23 Block IVPV . . . 484
3.24 Block MAM . . . 486
3.25 Block MGP . . . 488
3.26 Block NIEG . . . 490
3.27 Block PVA2C . . . 492
3.28 Block PVAI . . . 494
3.29 Block PVAV . . . 498
3.30 Block PVDET1 . . . 501
3.31 Block PVDET2 . . . 506
3.32 Block PVECON . . . 510
3.33 Block PVFIT1 . . . 513
3.34 Block PVFIT2 . . . 517
3.35 Block PVFITA . . . 521
3.36 Block PVI . . . 526
3.37 Block PVV . . . 532
3.38 Block STORE . . . 535
3.39 Block VBI . . . 538
3.40 Block XSNOFX . . . 540

4 Solar Thermal Energy 543
4.1 Block ACM . . . 545
4.2 Block AIR . . . 548
4.3 Block AIRHEAT . . . 550
4.4 Block COOLTW . . . 555
4.5 Block CTRHYS . . . 560
4.6 Block CTRL4DY . . . 562
4.7 Block CTRLDA . . . 564
4.8 Block DECCTL . . . 566
4.9 Block DECCTY . . . 569
4.10 Block DHYD . . . 571
4.11 Block DPBOWR . . . 573
4.12 Block DPBRAR . . . 575
4.13 Block DPCH . . . 577
4.14 Block DPDIF . . . 579
4.15 Block DPIN . . . 581
4.16 Block DPJUNR . . . 583
4.17 Block DPOUT . . . 585
4.18 Block DPPARA . . . 587
4.19 Block DPRED . . . 589
4.20 Block DPTUBE . . . 591
4.21 Block DPZETA . . . 593
4.22 Block DSTAR . . . 595
4.23 Block DWHEEL . . . 597

4.24 Block ESCOOL . . . 599
4.25 Block EVCOOL . . . 601
4.26 Block GR . . . 603
4.27 Block H2OHE . . . 605
4.28 Block H2OHL . . . 607
4.29 Block H2OHV . . . 609
4.30 Block HUMA2R . . . 611
4.31 Block HUMH . . . 613
4.32 Block HUMR2A . . . 615
4.33 Block HXC . . . 617
4.34 Block HXEARTH . . . 619
4.35 Block HXS . . . 622
4.36 Block HXXI . . . 624
4.37 Block LBWH . . . 627
4.38 Block LBWP . . . 629
4.39 Block LBWT . . . 631
4.40 Block LBWX . . . 633
4.41 Block MIXER . . . 635
4.42 Block MIXQHC . . . 637
4.43 Block PVT . . . 639
4.44 Block PVTCOOL . . . 642
4.45 Block R134A . . . 645
4.46 Block SATP . . . 646
4.47 Block SATT . . . 648
4.48 Block SCAIRC . . . 650
4.49 Block SCAIRCD . . . 653
4.50 Block SCDYN . . . 657
4.51 Block SCETA . . . 660
4.52 Block TANKFM . . . 663
4.53 Block TANKST . . . 665
4.54 Block THREEV . . . 668
4.55 Block TUBLO . . . 670
4.56 Block VBOWR . . . 671
4.57 Block VDIF . . . 673
4.58 Block VIN . . . 675
4.59 Block VOUT . . . 677
4.60 Block VRED . . . 679
4.61 Block VZETA . . . 681

5 Building Simulation 683
5.1 Block AMBIENT . . . 685
5.2 Block BASE . . . 687
5.3 Block CALEND . . . 689

5.4 Block CONZON . . . 690
5.5 Block CTRLROOM . . . 692
5.6 Block D18599 . . . 694
5.7 Block DAYLC . . . 698
5.8 Block DYBASE . . . 700
5.9 Block DYNDGL . . . 703
5.10 Block EINZON . . . 705
5.11 Block ENERBA . . . 708
5.12 Block F90 . . . 710
5.13 Block F90P . . . 711
5.14 Block FACADE . . . 713
5.15 Block FADYBS . . . 714
5.16 Block FDBS . . . 717
5.17 Block FIT3PC . . . 720
5.18 Block FIT3PH . . . 723
5.19 Block FIT4PC . . . 724
5.20 Block FIT4PH . . . 725
5.21 Block FIT5PH . . . 726
5.22 Block FPARA . . . 727
5.23 Block FPARAP . . . 728
5.24 Block GMIX . . . 729
5.25 Block HR . . . 731
5.26 Block IRADIA . . . 732
5.27 Block Q3PC . . . 733
5.28 Block Q3PH . . . 734
5.29 Block Q4PC . . . 735
5.30 Block Q4PH . . . 736
5.31 Block Q5P . . . 737
5.32 Block QDTV . . . 738
5.33 Block RADI . . . 739
5.34 Block ROOM . . . 741
5.35 Block SCHEDU . . . 743
5.36 Block WALL . . . 744
5.37 Block WALLX . . . 746
5.38 Block WINDOW . . . 748

6 Solar Power 751
6.1 Block CNTRL . . . 753
6.2 Block EVAPH . . . 754
6.3 Block FLDIV . . . 755
6.4 Block HXCH . . . 756
6.5 Block PCNTRL . . . 758
6.6 Block PCSSTU . . . 759

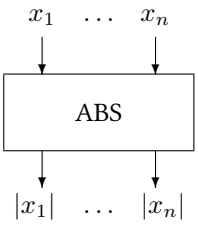
6.7 Block PFSTOR . . . 760
6.8 Block PIPE . . . 762
6.9 Block RECEIV . . . 763
6.10 Block SHADE . . . 765
6.11 Block STEAM . . . 766
6.12 Block STURB . . . 767
6.13 Block TC2TF . . . 768
6.14 Block TCSTOR . . . 769
6.15 Block TF2TC . . . 771
6.16 Block TPIECE . . . 772
6.17 Block TRACK . . . 773
6.18 Block TROUGH . . . 775

7 Community blocks 777
7.1 Block 2pcon2 . . . 779
7.2 Block UBBHKWSG . . . 780
7.3 Block UBBHKWWG . . . 782
7.4 Block UBH2COMPRESSOR . . . 784
7.5 Block UBH2CYLINDER . . . 785
7.6 Block UBISONLAND . . . 787
7.7 Block UBPEMECSTACK . . . 788
7.8 Block UBPEMFC . . . 790
7.9 Block UBSTORAGE . . . 792

1 :: Fundamental Blocks

1.1 Block ABS

The ABS block returns the absolute value of its input.



Name	ABS
Function	fb0005
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	0
Strings	0
Group	S

Inputs

i	Any value x_i

Outputs

i	Absolute value $\|x_i\|$ of input x_i

Parameters

None

Strings

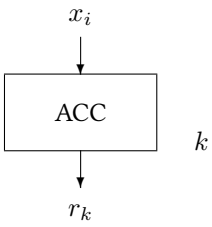
None

abs.vseit

A CONST block defines the parameter −17. The ABS block returns the absolute value of its input, here the output of the CONST block. The output 17 of the ABS block is then displayed using a SCREEN block.

1.2 Block ACC

The ACC block calculates the sample autocorrelation coefficient of its inputs at a given lag over a complete simulation run.



Name	ACC
Function	fb0052
Inputs	1
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	I

Inputs

1 Any signal x_i

Outputs

1 Autocorrelation coefficient r_k over all x_i

Parameters

1 Lag k (default $k = 1$)

Strings

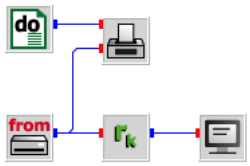
None

Description

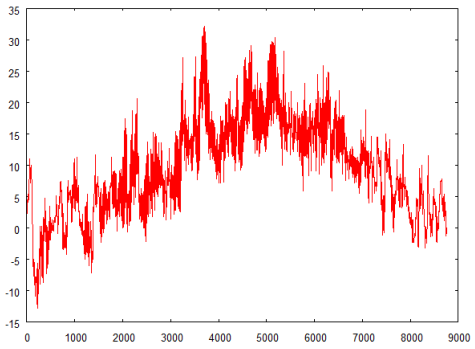
The auto-correlation coefficient r_k of a time series $\{x_i\}$ with lag k is given by

$$r_k = \frac{\sum_{i=1}^{N-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

acc.vseit



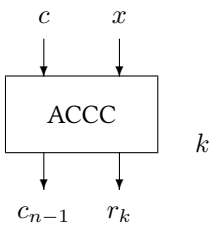
The hourly ambient temperature profile saved in `meteo82.dat` in the `examples/data` directory is read by a READ block for a complete year as defined through the DO block. A PLOT block is used to display the complete time series.



The ACC block calculates the auto-correlation coefficient for lag one, i. e., one hour. The annual value (0.994309 in this case) is displayed by the connected SCREEN block.

1.3 Block ACCC

The ACCC block calculates the sample autocorrelation coefficient at a given lag for constant conditions. When the condition input changes its value, the successors of the block are executed.



Name	ACCC
Function	fb0058
Inputs	2
Outputs	2
Parameters	0 . . . [1]
Strings	0
Group	I

Inputs

1 Condition c
2 Any signal x

Outputs

1 Condition c
2 Autocorrelation coefficient r_k over all calls with constant condition input.

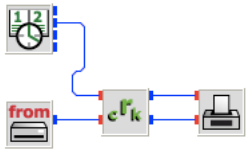
Parameters

1 Lag k (default $k = 1$)

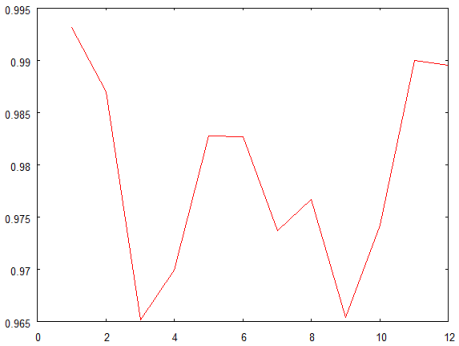
Strings

None

accc.vseit

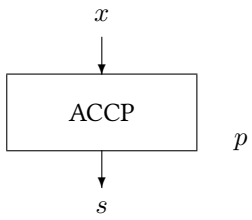


The calculation of the monthly auto-correlation coefficients is similar to the example for the ACC block. The same time series is used. Now the READ block is driven by a CLOCK block which delivers the condition input, i. e., the month signal to the ACCC block. The result is displayed by the PLOT block.



1.4 Block ACCP

The ACCP block calculates the sample autocorrelation coefficient at a given lag over a number of steps as defined through a parameter.



Name	ACCP
Function	fb0067
Inputs	1
Outputs	1
Parameters	1 . . . [2]
Strings	0
Group	I

Inputs

1 Any signal x

Outputs

1 Autocorrelation coefficient r_k over p calls.

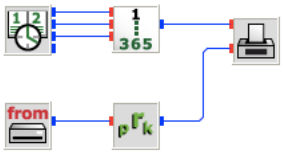
Parameters

1 Number of steps p
2 Lag k (default $k = 1$)

Strings

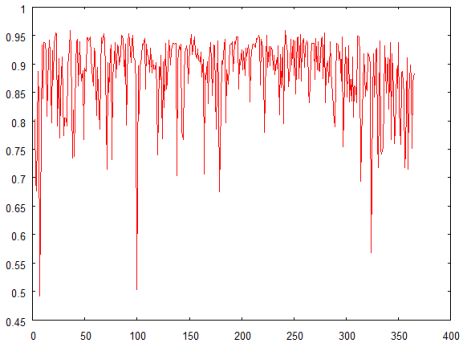
None

accp.vseit



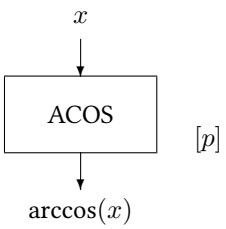
The calculation of the daily auto-correlation coefficients is similar to the examples for the ACC block and the ACCC block. Again, the same time series is used. The READ block is driven by a CLOCK block which delivers the year, month, day and hour in one-hour resolution.

The dates are converted into the day of the year by the DOY block as convenient x-axis for the PLOT block, which displays the time series of the hourly auto-correlation coefficients as calculated by the ACCP block.



1.5 Block ACOS

The ACOS block returns the arcus cosine of its input.



Name	ACOS
Function	fb0008
Inputs	1
Outputs	1
Parameters	$0 \ldots [1]$
Strings	0
Group	S

Inputs

1 Any value x with $|x| \leq 1$

Outputs

1 Arcus cosine of x

Parameters

1 Degrees / radians switch p. If p is set to 0 (default) $\arccos(x)$ is returned in degrees $(-90°, +90°)$. If p is set to 1 $\arccos(x)$ is returned in radians $(-\pi/2, +\pi/2)$.

Strings

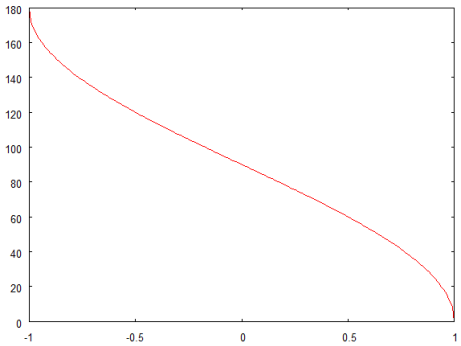
None

Remarks

If the amount of the input value $|x|$ exceeds one, the ACOS block displays a warning message and does no operation. If this happens more than once, the warning message is suppressed. At the end of a simulation run the total number of occurencies is displayed.

acos.vseit

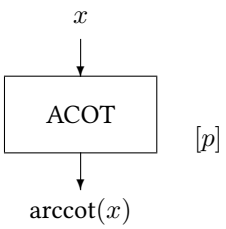
A DO block varies x from -1 to 1 in steps of 0.01. The ACOS block calculates the arc cosine of x and the PLOT block displays the graph on screen.



See also Blocks ASIN, ATAN, ACOT.

1.6 Block ACOT

The ACOT block returns the arcus cotangent of its input.



Name	ACOT
Function	fb0008
Inputs	1
Outputs	1
Parameters	$0 \ldots [1]$
Strings	0
Group	S

Inputs

1 Any value x

Outputs

1 Arcus cotangent of x

Parameters

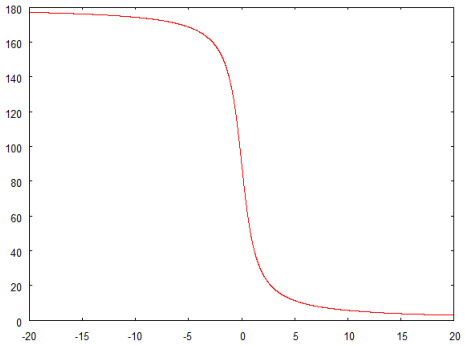
1 Degrees / radians switch p. If p is set to 0 (default) arccot(x) is returned in degrees $(-90°, +90°)$. If p is set to 1 arccot(x) is returned in radians $(-\pi/2, +\pi/2)$.

Strings

None

acot.vseit

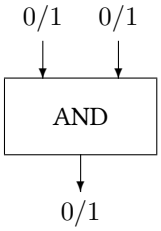
A DO block varies x from -10 to 10 in steps of 0.1. The ACOT block calculates the arc cotangent of x and the PLOT block displays the graph on screen.



See also Blocks ASIN, ACOS, ATAN.

1.7 Block AND

The AND block checks whether its first and second inputs are both equal to one (true).



Name	AND
Function	fb0030
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any value x_1 close to one or zero
2 Any value x_2 close to one or zero

Outputs

1 y is set to one if x_1 and x_2 are $\in [0.5, 1.5)$ otherwise y is set to zero

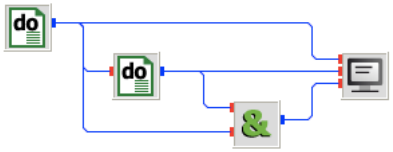
Parameters

None

Strings

None

and.vseit



Two nested DO blocks with initial value 0, final value 1 and increment 1 are linked to an AND block. The result is displayed by the SCREEN block.

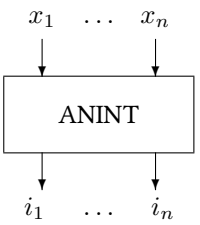
```
0.0   0.0   0.0
0.0   1.0   0.0
1.0   0.0   0.0
1.0   1.0   1.0
```

See also Blocks OR, XOR.

1.8 Block ANINT

The ANINT block returns the rounded integer value of the input signal.

12.34



Name	ANINT
Function	fb0005
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	0
Strings	0
Group	S

Inputs

i Any value r_i

Outputs

i Rounded value of x_i, ie if $n \leq x_i < n+1$ for any value $n \in \mathbf{N}$, rounded value is n if $x_i < n + 0.5$, $n + 1$ otherwise.

Parameters

None

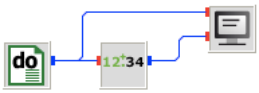
Strings

None

Remarks

The output of the ANINT block has a rounded value but is still a Fortran Real variable.

anint.vseit

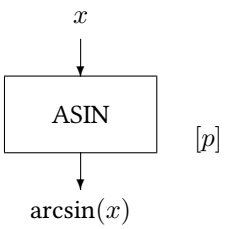


A DO block with initial value 0, final value 1 and increment 0.25 is linked to an ANINT block. The result is displayed by the SCREEN block.

```
0.00    0.00
0.25    0.00
0.50    1.00
0.75    1.00
1.00    1.00
```

1.9 Block ASIN

The ASIN block returns the arcus sine of its input.



Name	ASIN
Function	fb0008
Inputs	1
Outputs	1
Parameters	$0 \ldots [1]$
Strings	0
Group	S

Inputs

1 Any value x with $|x| \leq 1$

Outputs

1 Arcus sine of x

Parameters

1 Degrees / radians switch p. If p is set to 0 (default) $\arcsin(x)$ is returned in degrees $(-90°, +90°)$. If p is set to 1 $\arcsin(x)$ is returned in radians $(-\pi/2, +\pi/2)$.

Strings

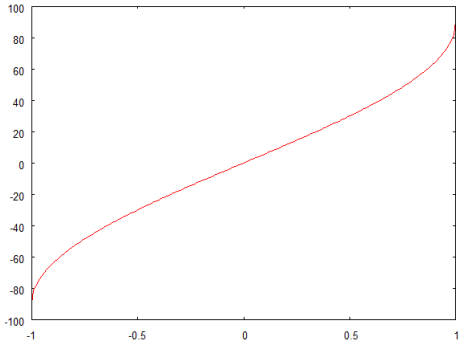
None

Remarks

If the amount of the input value $|x|$ exceeds one, the ASIN block displays a warning message and does no operation. If this happens more than once, the warning message is suppressed. At the end of a simulation run the total number of occurencies is displayed.

asin.vseit

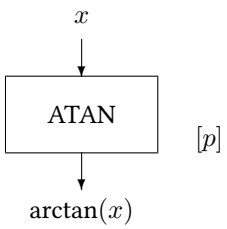
A DO block varies x from -1 to 1 in steps of 0.01. The ASIN block calculates the arc sine of x and the PLOT block displays the graph on screen.



See also Blocks ACOS, ATAN, ACOT.

1.10 Block ATAN

The ATAN block returns the arcus tangent of its input.



Name	ATAN
Function	fb0008
Inputs	1
Outputs	1
Parameters	$0 \ldots [1]$
Strings	0
Group	S

Inputs

1 Any value x

Outputs

1 Arcus tangent of x

Parameters

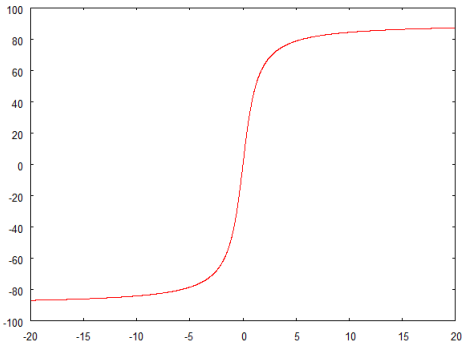
1 Degrees / radians switch p. If p is set to 0 (default) arctan(x) is returned in degrees $(-90°, +90°)$. If p is set to 1 arctan(x) is returned in radians $(-\pi/2, +\pi/2)$.

Strings

None

atan.vseit

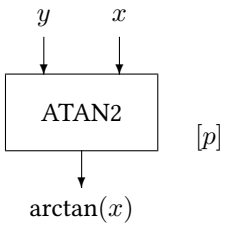
A DO block varies x from -10 to 10 in steps of 0.1. The ATAN block calculates the arc tangent of x and the PLOT block displays the graph on screen.



See also Blocks ASIN, ACOS, ACOT.

1.11 Block ATAN2

The ATAN2 block calculates the arctangent of the length ratio opposite leg divided by adjacent leg.



Name	ATAN2
Function	fb0069
Inputs	2
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Length of oppsite leg y
2 Length of adjacent leg x

Outputs

1 Arctangent of y/x

Parameters

None

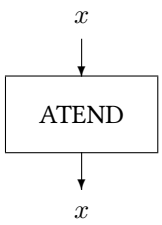
Strings

None

1.12 Block ATEND

The ATEND block executes its successors at the end of a simulation run only.

@end



Name	ATEND
Function	fb0045
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	0
Strings	0
Group	I

Inputs

1 Any signal x

Outputs

1 Signal x is transported only at the end of a simulation run

Parameters

None

Strings

None

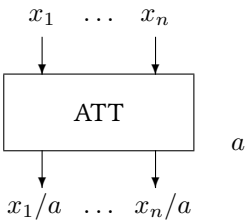
atend.vseit

In a DO block numbers are counted from 1 to 10. The result is filtered through the ATEND block. As a result the last output of the DO block is then displayed by the SCREEN block.

```
10.0 10.0
```

1.13 Block ATT

The ATT block divides its input by its parameter.



Name	ATT
Function	fb0006
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	1
Strings	0
Group	S

Inputs

i	Any value x_i

Outputs

i	Quotient of input and parameter x_i/a

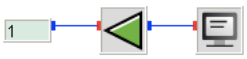
Parameters

1	Attenuation factor $a \neq 0$

Strings

None

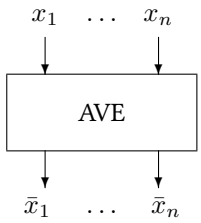
att.vseit



A CONST block with value 1 is used and attenuated by a factor 3.14. A SCREEN block displays the result 0.3185.

1.14 Block AVE

The AVE block calculates the average of its input signal over a complete simulation run.



Name	AVE
Function	fb0021
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	0
Strings	0
Group	I

Inputs

i Any signal x_i

Outputs

i Average $\bar{x}$ over all x_i

Parameters

None

Strings

None

ave.vseit

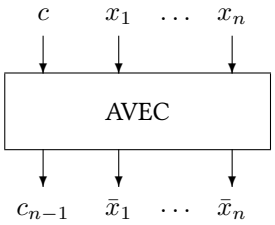
In a DO block numbers are counted from 1 to 10. The AVE block calculates the average over the complete run. The outputs of these two blocks are then displayed by the SCREEN block.

```
10.00    5.50
```

See also Blocks AVEC, AVEM, AVEP

1.15 Block AVEC

The AVEC block cumulates its second input signal as long as the condition input remains constant. When this input changes its value, the average is calculated and the successors of the block are executed.



Name	AVEC
Function	fb0037
Inputs	2 . . . [51]
Outputs	2 . . . [51]
Parameters	0
Strings	0
Group	I

Inputs

1	Condition input c
2	Any signal x_1
n	Any signal x_{n-1}

Outputs

1	Condition input c delayed by one step
2	Average $\bar{x}_1$
n	Average $\bar{x}_{n-1}$

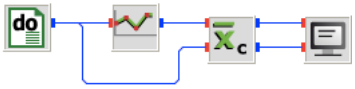
Parameters

None

Strings

None

avec.vseit



In a DO block numbers are counted from 1 to 6. The POLYG block provides the condition (either 1 or 2) as a function of the DO block's signal through the parameters

```
1 1
2 1
3 2
4 2
5 2
6 2
```

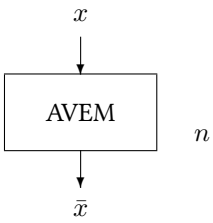
The AVEC block calculates the conditional average and finally the result is displayed by the SCREEN block.

```
1.   1.50
2.   4.50
```

See also Blocks AVE, AVEM, AVEP.

1.16 Block AVEM

The AVEM block calculates a moving average of its input signal over a given number of steps.



Name	AVEM
Function	fb0042
Inputs	1
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Any signal x

Outputs

1 Moving average $\bar{x}$ of the input signal over n steps

Parameters

1 Number n of steps over which the moving average is calculated

Strings

None

Description The output of this block is given by

$$\bar{x}_j = \frac{1}{\min\{n, j\}} \left(\sum_{i=\max\{1, j-n+1\}}^{j} x_i \right)$$

where

j	Actual timestep
n	Number of time steps over which the average is calculated

Thus the average is always calculated over the last n steps.

avem.vseit

In a DO block numbers are counted from 1 to 6. The AVEM block with its parameter set to 4 calculates the moving average and finally the results are displayed by the SCREEN block.

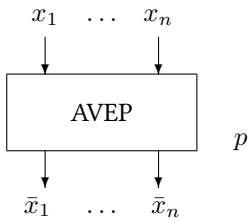
```
1.   1.00
2.   1.50
3.   2.00
4.   2.50
5.   3.50
6.   4.50
```

Please observe that the successors of the AVEM block are executed in every time step (in contrast to the AVE block, for instance).

See also Blocks AVE, AVEC, AVEP.

1.17 Block AVEP

The AVEP block calculates the average of its input signal over a number of steps as defined through its parameter.



Name	AVEP
Function	fb0020
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	1
Strings	0
Group	I

Inputs

i Any signal x_i

Outputs

i Average $\bar{x}$

Parameters

1 Number p of steps to build the average $\bar{x}$

Strings

None

avep.vseit

In a DO block numbers are counted from 1 to 6. The AVEP block calculates the average over every two steps and the results are displayed by the SCREEN block.

```
2.   1.50
4.   3.50
6.   5.50
```

See also Blocks AVE, AVEC, AVEM.

1.18 Block AVEW

The AVEW block calculates the weighted average of its inputs.

Name	AVEW
Function	fb0078
Inputs	2
Outputs	1
Parameters	2
Strings	0
Group	S

Inputs

1 Any signal T_1
2 Any signal T_2

Outputs

1 Weighted average $barT_{\mathrm{w}}$

Parameters

1 Weight one w_1
2 Weight two w_2

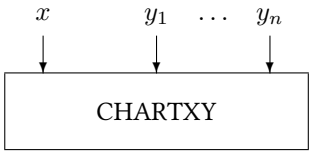
Strings

None

1.19 Block CHARTXY

Hello Kai! Write LaTeXdocumentation here. and here and here.

CHARTXY is a Java block, and does not display anything when run from the console. Use PLOT instead if you want to create diagrams from the console.



Name	CHARTXY
Function	fb0000
Inputs	1 . . . [100]
Outputs	0
Parameters	0
Strings	0
Group	S

Inputs

1	Any signal x
2	Any signal y_1
n	Any signal y_{n-1}

Outputs

None

Parameters

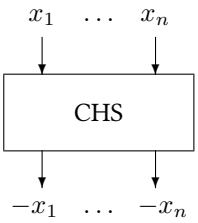
None

Strings

None

1.20 Block CHS

The CHS block changes the sign of its input signal.



Name	CHS
Function	fb0005
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	0
Strings	0
Group	S

Inputs

i Any value x_i

Outputs

i Inversion $-x_i$

Parameters

None

Strings

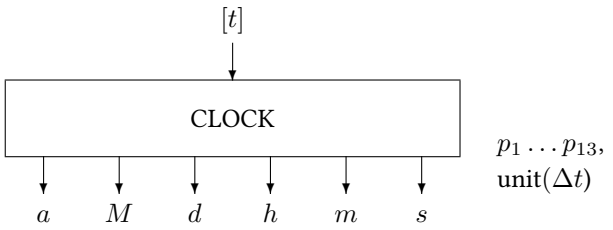
None

chs.vseit

A CONST block defines the parameter 17. The block CHS changes the sign of its input, here the output of the CONST block. The output −17 of the CHS block is then displayed using the SCREEN block.

1.21 Block CLOCK

The CLOCK block generates date and time of the actual simulation time step with constant increment.



Name	CLOCK
Function	fb0024
Inputs	0 . . . [1]
Outputs	6
Parameters	13
Strings	1
Group	T

Inputs

1 Output t of a predecessor (optional). Should the block defining the t signal be executed again after CLOCK has finished its operation, the CLOCK block performs a reset and starts again.

Outputs

1 Year a
2 Month M
3 Day d
4 Hour h
5 Minute m
6 Second s

Parameters

1 Start on year a_1
2 Start on month M_1
3 Start on day d_1
4 Start on hour h_1
5 Start on minute m_1

6 Start on second s_1
7 Stop on year a_2
8 Stop on month M_2
9 Stop on day d_2
10 Stop on hour h_2
11 Stop on minute m_2
12 Stop on second s_2
13 Increment Δt

Strings

1 Unit of the increment Δt, case sensitive, ie 'm' $\neq$ 'M', for example
'a' Years
'M' Months
'd' Days
'h' Hours
'm' Minutes
's' Seconds

---

clock.vseit

The CLOCK block runs through two hours with a time step of 20 minutes. It starts on the February 28 at 23 o'clock and ends on Februray 29 at 1 o'clock. The result (year a, month M, day d, hour h, minute m and second s) is shown by a SCREEN block.

```
2012.    2.   28.   23.    0.    0.
2012.    2.   28.   23.   20.    0.
2012.    2.   28.   23.   40.    0.
2012.    2.   29.    0.    0.    0.
2012.    2.   29.    0.   20.    0.
2012.    2.   29.    0.   40.    0.
```

Remarks (1) Please note how the CLOCK block handles the time variables. The time, which is an output of the CLOCK block, remains constant during the whole time step, and is set to the start value of each time interval. The stop time 01:00:00 is given as a block parameter. The simulation time given by the start and stop time parameters is two hours in total. With a time step of 20 minutes this parameter setting results in six time steps, so that the CLOCK block starts at 23:00:00 and ends at 00:40:00—six time steps later.

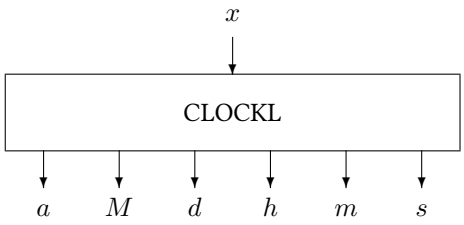
(2) Please observe that the stop time at 1 o'clock does not appear as output, since this would indicate a seventh time step and this would be wrong!

(3) The CLOCK block handles midnight of a day—28 February 2000, for instance—as time 00:00:00 of the following day.

See also Blocks GREGOR, JULIAN, DOW.

1.22 Block CLOCKL

The CLOCKL block is an L-block version of the CLOCK block..



Name	CLOCKL
Function	fb0072
Inputs	1
Outputs	6
Parameters	13
Strings	1
Group	L

Inputs

1 Output of a predecessor.

Outputs

1 Year a
2 Month M
3 Day d
4 Hour h
5 Minute m
6 Second s

Parameters

1 Start on year a_1
2 Start on month M_1
3 Start on day d_1
4 Start on hour h_1
5 Start on minute m_1
6 Start on second s_1
7 Stop on year a_2
8 Stop on month M_2

9	Stop on day d_2
10	Stop on hour h_2
11	Stop on minute m_2
12	Stop on second s_2
13	Increment Δt

Strings

1	Unit of the increment Δt, case sensitive, ie 'm' $\neq$ 'M', for example
	'a' Years
	'M' Months
	'd' Days
	'h' Hours
	'm' Minutes
	's' Seconds

---

clockl.vseit

The CLOCKL block runs through two hours with a time step of 20 minutes. It starts on the February 28 at 23 o'clock and ends on Februray 29 at 1 o'clock. The result (year a, month M, day d, hour h, minute m and second s) is shown by a SCREEN block.

The DO block runs the loop two times.

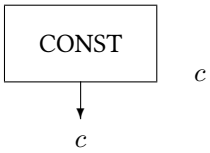
```
2012.    2.   28.   23.    0.    0.
2012.    2.   28.   23.   20.    0.
2012.    2.   28.   23.   40.    0.
2012.    2.   29.    0.    0.    0.
2012.    2.   29.    0.   20.    0.
2012.    2.   29.    0.   40.    0.
2012.    2.   28.   23.    0.    0.
2012.    2.   28.   23.   20.    0.
2012.    2.   28.   23.   40.    0.
2012.    2.   29.    0.    0.    0.
2012.    2.   29.    0.   20.    0.
2012.    2.   29.    0.   40.    0.
```

Remarks Please notice the required connection to the TOL block and the feedback loop.

See also Blocks CLOCK.

1.23 Block CONST

The CONST block defines a constant.



Name	CONST
Function	fb0001
Inputs	0
Outputs	1
Parameters	1
Strings	0
Group	C

Inputs

None

Outputs

1 Constant c

Parameters

1 Constant c

Strings

None

const.vseit

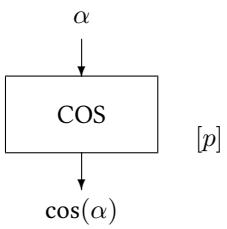
A CONST block provides the value 3.14 which is displayed by the SCREEN block as

```
3.1400001
```

This is due to the used star format of the SCREEN block and the inherent accuracy of Fortran Real variables.

1.24 Block COS

The COS block returns the cosine of its input.



Name	COS
Function	fb0007
Inputs	1
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Any angle α

Outputs

1 Cosine of α

Parameters

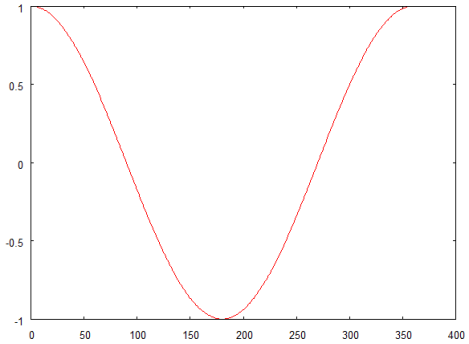
1 Degrees / radians switch p. If p is set to 0 (default) α is interpreted in degrees. If p is set to 1 α is interpreted in radians.

Strings

None

cos.vseit

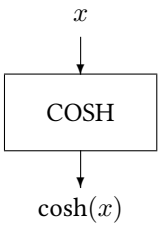
A DO block varies α from 0 to 360 degrees. The COS block calculates $\cos(\alpha)$ and the PLOT block displays the graph on screen.



See also Blocks ACOS, ACOT, ASIN, ATAN, COT, SIN, and TAN.

1.25 Block COSH

The COSH block returns the hyperbolic cosine of its input.



Name	COSH
Function	fb0068
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any signal x

Outputs

1 Hyperbolic cosine

Parameters

None

Strings

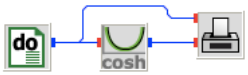
None

Description

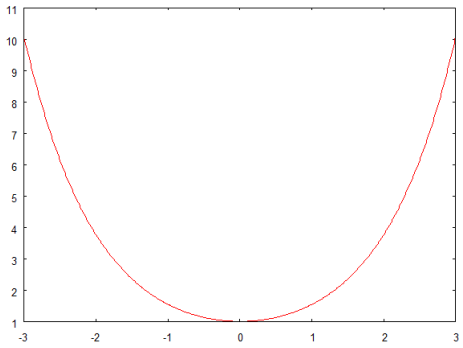
The hyperbolic cosine is defined as

$$\cosh(x) = \frac{\exp(x) + \exp(-x)}{2}$$

cosh.vseit



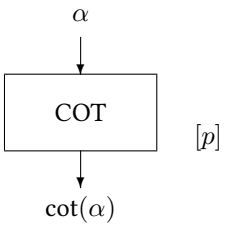
A DO block varies x from -3 to 3 in steps of 0.01. The COSH block calculates $\cosh(x)$ and the PLOT block displays the graph on screen.



See also Blocks COTH, CSCH, SCH, SINH, TANH.

1.26 Block COT

The COT block returns the cotangent of its input.



Name	COT
Function	fb0007
Inputs	1
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Any angle α

Outputs

1 Cotangent of α

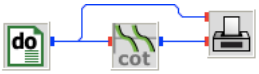
Parameters

1 Degrees / radians switch p. If p is set to 0 (default) α is interpreted in degrees. If p is set to 1 α is interpreted in radians.

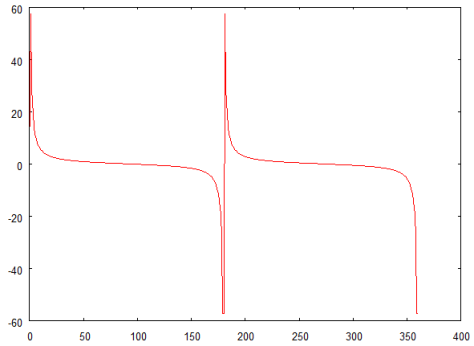
Strings

None

cot.vseit



A DO block varies α from 0 to 360 degrees in steps of one degree. The COT block calculates $\cot(\alpha)$ and the PLOT block displays the graph on screen.



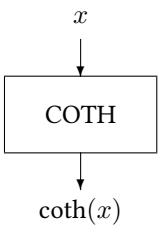
The COT block recognizes the discontinuities at 0, 180, and 360 degrees and displays warnings.

```
W05007 Block 00002: Input angle too close to discontinuity
W05008 Block 00002: Calls close to discontinuity:        3
```

See also Blocks ACOS, ACOT, ASIN, ATAN, COS, SIN, and TAN.

1.27 Block COTH

The COTH block returns the hyperbolic cotangent of its input.



Name	COTH
Function	fb0068
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any signal x

Outputs

1 Hyperbolic cotangent

Parameters

None

Strings

None

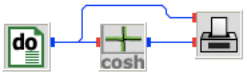
Description

The hyperbolic cotangent is defined as

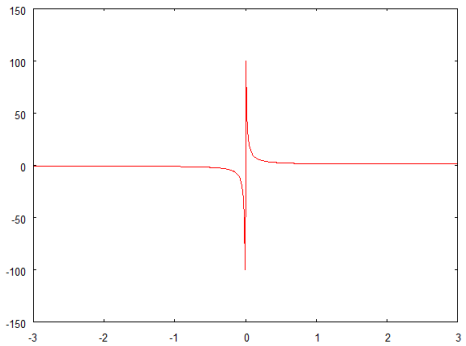
$$\coth(x) = \frac{\exp(x) + \exp(-x)}{\exp(x) - \exp(-x)}$$

In order to avoid division by zero, $\coth(x)$ is set to zero for $|x| < 10^{-5}$.

coth.vseit



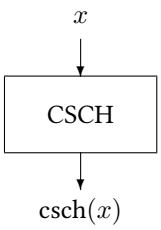
A DO block varies x from -3 to 3 in steps of 0.01. The COTH block calculates $\coth(x)$ and the PLOT block displays the graph on screen.



See also Blocks COSH, CSCH, SCH, SINH, TANH.

1.28 Block CSCH

The CSCH block returns the hyperbolic cosecant of its input.



Name	CSCH
Function	fb0068
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any signal x

Outputs

1 Hyperbolic cosecant

Parameters

None

Strings

None

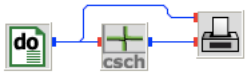
Description

The hyperbolic cosecant is defined as

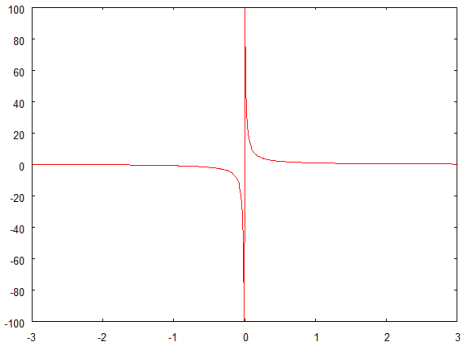
$$\operatorname{csch}(x) = \frac{2}{\exp(x) - \exp(-x)}$$

In order to avoid division by zero, $\operatorname{csch}(x)$ is set to zero for $|x| < 10^{-5}$.

csch.vseit



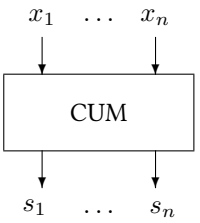
A DO block varies x from -3 to 3 in steps of 0.01. The CSCH block calculates csch(x) and the PLOT block displays the graph on screen.



See also Blocks COSH, COTH, SCH, SINH, TANH.

1.29 Block CUM

The CUM block cumulates its input signal over a complete simulation run.



Name	CUM
Function	fb0021
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	0
Strings	0
Group	I

Inputs

i Any signal x_i

Outputs

i Cumulated sum s_i over all x_i

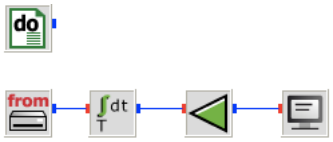
Parameters

None

Strings

None

cum.vseit



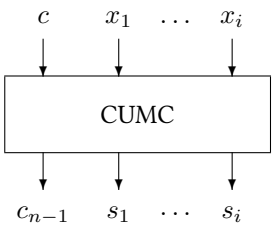
A DO block runs through all 8760 hours of a year. The READ block reads radiation data from a data file in W/m^2. The CUM block cumulates the data over the complete run. An ATT block converts the result into kWh/m^2. Finally, the SCREEN block displays the result on screen.

```
Annual cumulated radiation 954.69 kWh/m^2
```

See also Blocks CUMC and CUMP.

1.30 Block CUMC

The CUMC block cumulates its second input signal as long as the condition input remains constant. When this input changes its value, the cumulated sum is on output and the successors of the block are executed.



Name	CUMC
Function	fb0037
Inputs	2 . . . [51]
Outputs	2 . . . [51]
Parameters	0
Strings	0
Group	I

Inputs

1	Condition input c
2	Any signal x_1
n	Any signal x_{n-1}

Outputs

1	Condition input c delayed by one step
2	Cumulated sum s_1 over all $c =$ const calls
n	Cumulated sum s_{n-1} over all $c =$ const calls

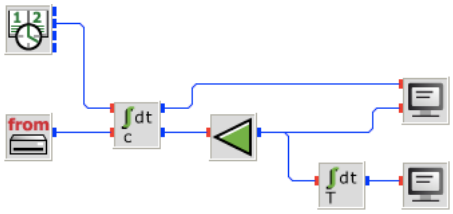
Parameters

None

Strings

None

cumc.vseit



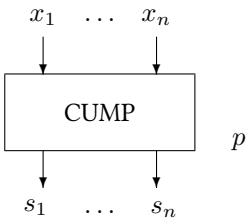
A CLOCK block runs through all 8760 hours of a year. The READ block reads radiation data from a data file in W/m^2. The CUMC block cumulates the data over each month. An ATT block converts the result into kWh/m^2. Finally, the SCREEN block displays the result on screen. In addition, the CUM cumulates the monthly values over the whole year and a second SCREEN block displays the annual result.

```
Month  1.: Radiation    21.86 kWh/m^2
Month  2.: Radiation    33.94 kWh/m^2
Month  3.: Radiation    74.66 kWh/m^2
Month  4.: Radiation   119.25 kWh/m^2
Month  5.: Radiation   142.43 kWh/m^2
Month  6.: Radiation   144.02 kWh/m^2
Month  7.: Radiation   160.57 kWh/m^2
Month  8.: Radiation   105.14 kWh/m^2
Month  9.: Radiation    84.20 kWh/m^2
Month 10.: Radiation    37.78 kWh/m^2
Month 11.: Radiation    18.99 kWh/m^2
Month 12.: Radiation    11.85 kWh/m^2
Annual radiation   954.69 kWh/m^2
```

See also Blocks CUM, CUMP.

1.31 Block CUMP

The CUMP block cumulates its input signal over a number of steps as defined through its parameter.



Name	CUMP
Function	fb0020
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	1
Strings	0
Group	I

Inputs

i Any signal x_i

Outputs

i Cumulated sum s_i of x_i over p calls

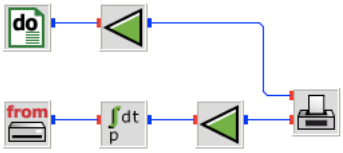
Parameters

1 Number p of steps to cumulate the input x_i

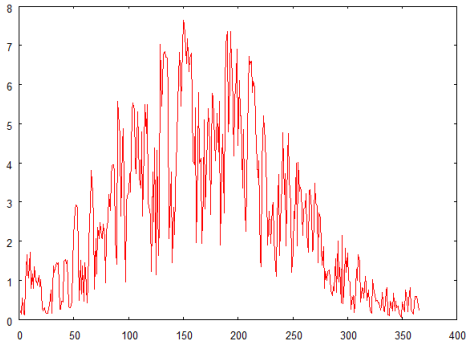
Strings

None

cump.vseit



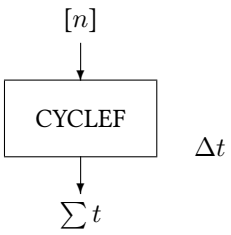
A DO block runs through all 8760 hours of a year. The READ block reads radiation data from a data file in W/m^2. The CUMP block cumulates the data over the day. An ATT block converts the result into kWh/m^2. Finally, the PLOT block displays the time series of daily radiation data.



See also Blocks CUM, CUMC.

1.32 Block CYCLEF

The CYCLEF block suspends execution of a simulation model for an interval specified in seconds.



Name	CYCLEF
Function	fb0051
Inputs	0 . . . [1]
Outputs	1
Parameters	1
Strings	0
Group	T

Inputs

1 Output n of a predecessor (optional)

Outputs

1 Cumulated time $\sum t$ / s

Parameters

1 Sleeping time Δt / s

Strings

None

cyclef.vseit

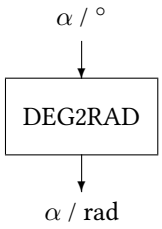
The CYCLEF block with its parameter set to 1 executes the model once every second. In the meantime the INSEL model is idle. The SCREEN block displays the cumulated time.

```
0.0000000
1.0000000
2.0000000
3.0000000
4.0000000
```

Model execution can be stopped via the Stop button or via Control-C if executed in a DOS box or terminal.

1.33 Block DEG2RAD

The DEG2RAD block changes degrees to radians.



Name	DEG2RAD
Function	fb0009
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any angle α in degrees

Outputs

1 α in radians

Parameters

None

Strings

None

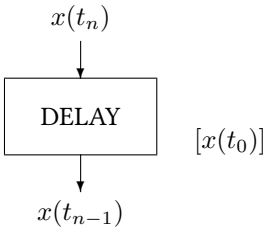
deg2rad.vseit

A CONST block provides the value 180 degrees which is converted to radians by the DEG2RAD block to

```
3.1415927
```

1.34 Block DELAY

The DELAY block delays its input signal by a single step.



Name	DELAY
Function	fb0015
Inputs	1 . . . [1000]
Outputs	10
Parameters	0 . . . [1]
Strings	0
Group	D

Inputs

1	Any value $x_1(t_n)$
2	Any value $x_2(t_n)$
i	Any value $x_i(t_n)$

Outputs

1	Delayed signal $x_1(t_{n-1})$
2	Delayed signal $x_2(t_{n-1})$
i	Delayed signal $x_i(t_{n-1})$

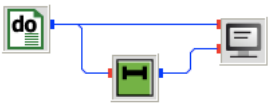
Parameters

1	Initial value $x_1(t_0)$ set to 0 by default.
2	Initial value $x_2(t_0)$ set to 0 by default.
i	Initial value $x_i(t_0)$ set to 0 by default.

Strings

None

delay.vseit

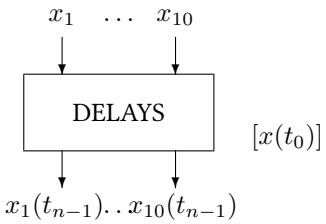


A DO block counts from 1 to 5. Its output is connected to a DELAY block with initial value zero. The outputs are displayed by the SCREEN block as

```
1.0   0.0
2.0   1.0
3.0   2.0
4.0   3.0
5.0   4.0
```

1.35 Block DELAYS

The DELAYS block delays its input signal by a single step.



Name	DELAYS
Function	fb0006
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	1
Strings	0
Group	S

Inputs

i	Any value $x_i(t_n)$

Outputs

i	Delayed signal $x_i(t_{n-1})$

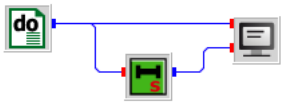
Parameters

1	Initial value $x_i(t_0)$

Strings

None

delays.vseit



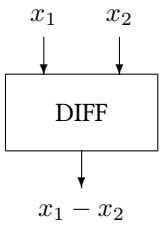
A DO block counts from 1 to 5. Its output is connected to a DELAYS block with initial value zero. The outputs are displayed by the SCREEN block as

```
1.0   0.0
2.0   1.0
3.0   2.0
4.0   3.0
5.0   4.0
```

Please observe that—in contrast to the DELAY block—the DELAYS block cannot be used to solve algebraic loops.

1.36 Block DIFF

The DIFF block expects exactly 2 signals, x1 and x2, and returns x1 - x2.



Name	DIFF
Function	fb0084
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Minuend
2	Subtrahend

Outputs

1	Difference

Parameters

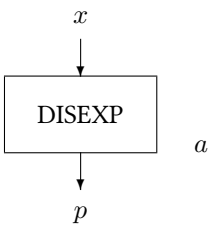
None

Strings

None

1.37 Block DISEXP

The DISEXP block calculates the exponential distribution.



Name	DISEXP
Function	fb0029
Inputs	1
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Independent variable x

Outputs

1 Probability density $p(x)$

Parameters

1 Distribution parameter a

Strings

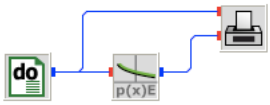
None

Description

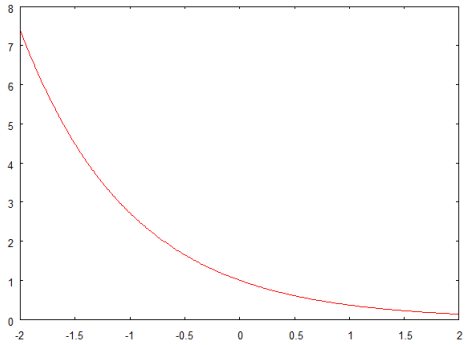
The exponential distribution is defined by the probability density function

$$p(x) = \frac{1}{a} \exp\left(\frac{-x}{a}\right)$$

disexp.vseit

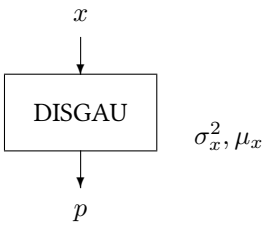


A DO block varies x from from -2 to 2 in steps of 0.01. The DISEXP block with its parameter set to one calculates the exponential distribution which is plotted by a PLOT block.



1.38 Block DISGAU

The DISGAU block calculates the normal distribution.



Name	DISGAU
Function	fb0029
Inputs	1
Outputs	1
Parameters	2
Strings	0
Group	S

Inputs

1 Independent variable x

Outputs

1 Probability density $p(x)$

Parameters

1 Variance σ_x^2
2 Mean value μ_x

Strings

None

Description

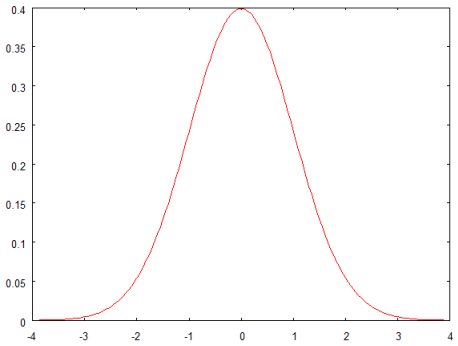
The probability density function

$$p(x) = \frac{1}{\sigma_x \sqrt{2\pi}} \exp\left(\frac{-(x-\mu_x)^2}{2\sigma_x^2}\right)$$

is called normal distribution or Gauß distribution after the German mathematician Karl Friedrich Gauß (1777–1855).

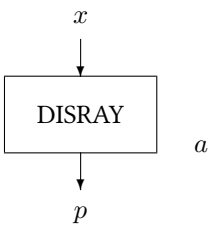
disgau.vseit

A DO block varies x from -4 to 4 in steps of 0.05. The DISGAU block with mean value zero and variance one calculates the normal distribution which is plotted by a PLOT block.



1.39 Block DISRAY

The DISRAY block calculates the Rayleigh distribution.



Name	DISRAY
Function	fb0029
Inputs	1
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Independent variable x

Outputs

1 Probability density $p(x)$

Parameters

1 Distribution parameter a

Strings

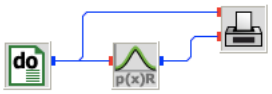
None

Description

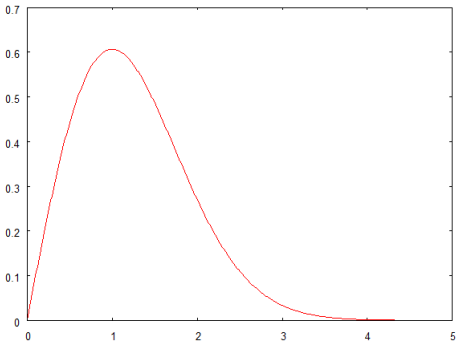
The Rayleigh distribution is defined by the probability density function

$$p(x) = \max\left\{0, \frac{x}{a^2}\exp\left(\frac{-x^2}{2a^2}\right)\right\}$$

disray.vseit

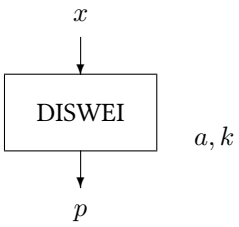


A DO block varies x from 0 to 5 in steps of 0.01. The DISRAY block calculates the Rayleigh distribution with distribution parameter one which is plotted by a PLOT block.



1.40 Block DISWEI

The DISWEI block calculates the Weibull distribution.



Name	DISWEI
Function	fb0029
Inputs	1
Outputs	1
Parameters	2
Strings	0
Group	S

Inputs

1 Independent variable x

Outputs

1 Probability density $p(x)$

Parameters

1 Scale parameter a
2 Shape parameter k

Strings

None

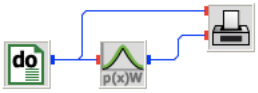
Description

The Weibull distribution is defined by the probability density function

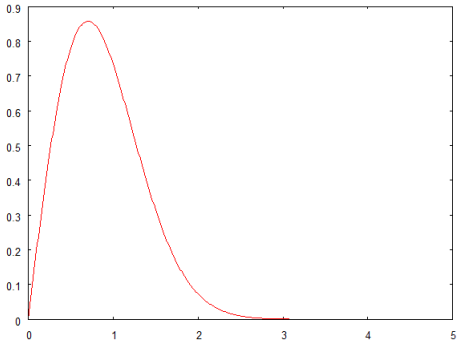
$$p(x) = \frac{k}{a}\left(\frac{x}{a}\right)^{k-1} \exp\left(-\frac{x^k}{a^k}\right) \tag{1.1}$$

For $k = 2$ the Weibull distribution degenerates to the Rayleigh distribution, and for $k = 1$ equals the exponential distribution.

diswei.vseit

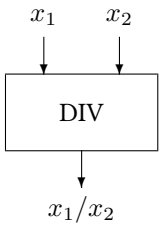


A DO block varies x from 0 to 5 in steps of 0.01. The DISWEI block calculates the Weibull distribution with scale parameter one and shape parameter two which is plotted by a PLOT block.



1.41 Block DIV

The DIV block divides its first input by the second.



Name	DIV
Function	fb0004
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Numerator x_1 of the fraction
2 Denominator x_2 of the fraction

Outputs

1 x_1 divided by x_2

Parameters

None

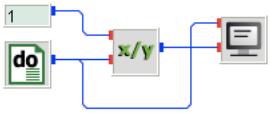
Strings

None

Remarks

If the second input value x_2 is zero, the block displays a warning message and returns NaN. If this happens more than once, the warning message is suppressed. At the end of a simulation run the total number of occurencies is displayed.

div.vseit

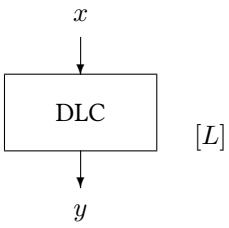


In a DO block numbers are counted from −3 to +3. A CONST block provides a constant set to 1. The DIV block divides 1 by the output of the DO block. The results are displayed using the SCREEN block.

```
  -3.000  -0.333
  -2.000  -0.500
  -1.000  -1.000
W05001 Block 00002: Division by zero
   0.000  -1.000
   1.000   1.000
   2.000   0.500
   3.000   0.333
W05002 Block 00002: Number of divisions by zero:        1
```

1.42 Block DLC

The DLC block tests whether the input signal is downward crossing a threshold level.



Name	DLC
Function	fb0039
Inputs	1
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Any input signal x.

Outputs

1 Indicator y for downward level crossing, y is set to one if x has crossed L from $x > L$, else y set to zero

Parameters

1 Threshold level L (default $L = 0$)

Strings

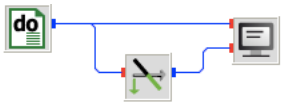
None

Description

The output of this block is defined as

$$y = \begin{cases} 1 & \text{if } x \text{ has crossed } L \text{ from } x > L \text{ to } x \leq L \\ 0 & \text{else} \end{cases}$$

dlc.vseit



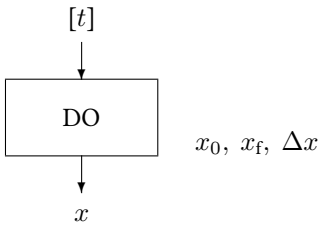
A DO block decreases numbers from 5 to 1 in steps of −1. The DLC block checks for downward level crossing at threshold 2.5. Both the counter and the output of the DLC block are then displayed using the SCREEN block.

```
5.   0.
4.   0.
3.   0.
2.   1.
1.   0.
```

See also Block ULC.

1.43 Block DO

The DO block returns numerical values with a constant increment.



Name	DO
Function	fb0013
Inputs	0 . . . [1]
Outputs	1
Parameters	3
Strings	0
Group	T

Inputs

1 Output t of a predecessor (optional). Should the block defining the t signal be executed again after DO has finished its operation, the DO block performs a reset and starts again.

Outputs

1 The output signal x runs from x_0 to x_f in steps of Δx

Parameters

1 Initial value x_0
2 Final value x_f
3 Increment Δx

Strings

None

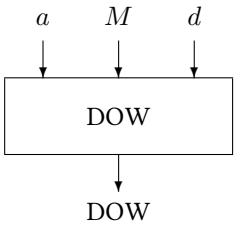
do.vseit

DO blocks can be nested. The outer DO block (which has no input) counts from 1 to 3. The second DO block uses the output of the outer DO block and varies its output from 0 to 10 in steps of 5. The outputs are displayed by a SCREEN block.

```
1.0     0.0
1.0     5.0
1.0    10.0
2.0     0.0
2.0     5.0
2.0    10.0
3.0     0.0
3.0     5.0
3.0    10.0
```

1.44 Block DOW

The DOW block returns the day of the week for a given Gregorian Date.



Name	DOW
Function	fb0025
Inputs	3
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Year a
2 Month M
3 Day d

Outputs

1 Day of week d_w
- 1 Monday
- 2 Tuesday
- 3 Wednesday
- 4 Thursday
- 5 Friday
- 6 Saturday
- 7 Sunday

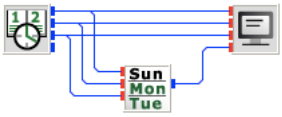
Parameters

None

Strings

None

dow.vseit

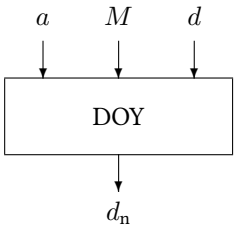


A CLOCK block counts the first seven days of January 2010. The DOW block returns the day of the week which is displayed by the SCREEN block.

```
2010.    1.    1.    5.
2010.    1.    2.    6.
2010.    1.    3.    7.
2010.    1.    4.    1.
2010.    1.    5.    2.
2010.    1.    6.    3.
2010.    1.    7.    4.
```

1.45 Block DOY

The DOY block returns the day of the year.



Name	DOY
Function	fb0023
Inputs	3
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Year a
2 Month M
3 Day d

Outputs

1 Day of the year $d_n \in [1, 366]$

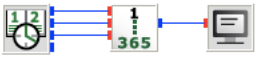
Parameters

None

Strings

None

doy.vseit

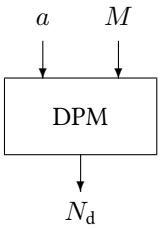


A CLOCK block provides the dates of the last four days of the year 2012 (a leap year). The DOY year block calculates the corresponding day of the year. The SCREEN block displays the result.

```
363.
364.
365.
366.
```

1.46 Block DPM

The DPM block returns the number of days in a month.



Name	DPM
Function	fb0066
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Year, e. g., 2008
2 Month

Outputs

1 N_d Number of days

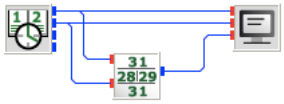
Parameters

None

Strings

None

dpm.vseit

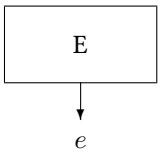


A CLOCK block provides the year and months of the year 2012 (a leap year) in steps of one month. The DPM block calculates the number of days of the respective months. The SCREEN block displays the result.

```
2012.    1.   31.
2012.    2.   29.
2012.    3.   31.
2012.    4.   30.
2012.    5.   31.
2012.    6.   30.
2012.    7.   31.
2012.    8.   31.
2012.    9.   30.
2012.   10.   31.
2012.   11.   30.
2012.   12.   31.
```

1.47 Block E

The E block provides the Euler constant.



Name	E
Function	fb0003
Inputs	0
Outputs	1
Parameters	0
Strings	0
Group	C

Inputs

None

Outputs

1 Euler Constant $e = \exp(1)$

Parameters

None

Strings

None

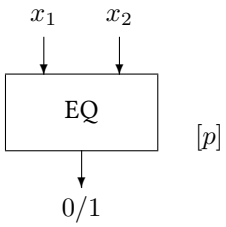
e.vseit

The E block provides the Euler constant which is displayed by the SCREEN block.

```
2.7182817
```

1.48 Block EQ

The EQ block tests whether its first input is equal to the second one.



Name	EQ
Function	fb0027
Inputs	2
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Any value x_1
2 Any value x_2

Outputs

1 y is set to one if $|x_1 - x_2| \leq p$, otherwise y is set to zero

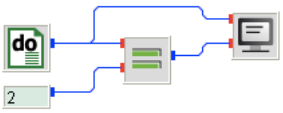
Parameters

1 Error tolerance $p \geq 0$ for the difference between x_1 and x_2, set to zero by default

Strings

None

eq.vseit

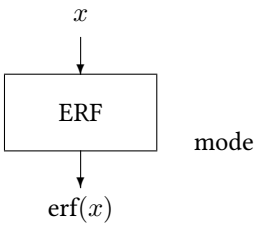


A DO block counts from 1 to 3. The CONST block provides a constant with value 2. The EQ blocks checks, whether the output of the DO block is equal to 2 (true = 1) or not (false = 0). The SCREEN block displays the result.

```
1.0   0.0
2.0   1.0
3.0   0.0
```

1.49 Block ERF

The ERF block calculates the error function.



Name	ERF
Function	fb0040
Inputs	1
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Any input signal x.

Outputs

1 Error function erf(x).

Parameters

1 Mode
- 0 Polynom of order three
- 1 Polynom of order five

Strings

None

Description In mode 0 the ERF block uses the third order polynom

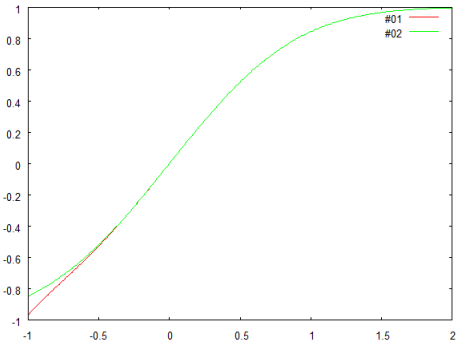
$$\begin{aligned} z &= \frac{1.0}{1.0 + 0.47047\,x} \\ \mathrm{erf}(x) &= 1.0 - (0.3480242\,z - 0.0958798\,z^2 \\ &\quad + 0.7478556\,z^3)\exp(-x^2) \end{aligned}$$

In mode 1 the ERF block uses the fifth order polynom

$$\begin{aligned} z &= \frac{1.0}{1.0 + 0.3275911x} \\ \mathrm{erf}(x) &= 1.0 - (0.254829592\,z - 0.284496736\,z^2 \\ &\quad + 1.421413741\,z^3 - 1.453152027\,z^4 \\ &\quad + 1.061405429\,z^5)\exp(-x^2) \end{aligned}$$

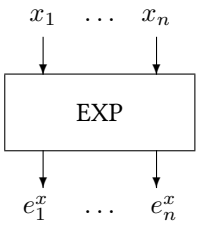
erf.vseit

A DO block varies x from -1 to 1 in steps of 0.1. The first ERF block calculates the error function by a polynom of order three (mode 0) and the second one by a polynom of order five (mode 1). The PLOT block displays both graphs on screen with mode 0 in red.



1.50 Block EXP

The EXP block defines the exponential function.



Name	EXP
Function	fb0005
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	0
Strings	0
Group	S

Inputs

i	Any value x_i

Outputs

i	$\exp(x_i) = e_i^x$

Parameters

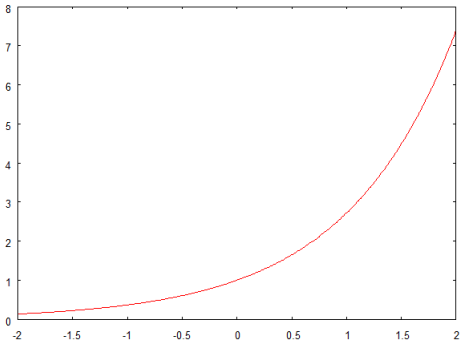
None

Strings

None

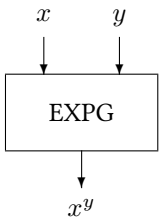
exp.vseit

A DO block varies x from -2 to 2 in steps of 0.01. The EXP block calculates the exponential function $\exp(x)$ and the PLOT block displays the graph on screen.



1.51 Block EXPG

The EXPG block defines a general exponential function.



Name	EXPG
Function	fb0010
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any basis value $x > 0$
2 Any exponent value y

Outputs

1 x^y

Parameters

None

Strings

None

Remarks

If the first input value x is negative, the EXPG block displays a warning message and does no operation. If this happens more than once, the warning message is suppressed. At the end of a simulation run the total number of occurencies is displayed.

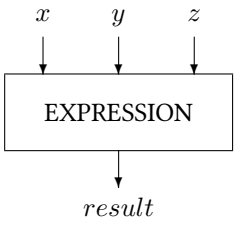
expg.vseit

Two CONST blocks provide the constants two and three. The EXPG block calculates 2^3 and the SCREEN block displays the result

```
8.0
```

1.52 Block EXPRESSION

Evaluates a mathematical expression, with inputs as x, y, z. E.g : x + y * z (sqrt(5) + 1) / 2 sin(2 * pi * x)



Name	EXPRESSION
Function	fb0086
Inputs	1
Outputs	1
Parameters	0
Strings	1
Group	S

Inputs

1	x
2	y
3	z

Outputs

1	Result

Parameters

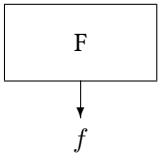
None

Strings

1	Expression

1.53 Block F

The F block provides the *dilution factor*: the ratio of irradiances between the solar constant on Earth, and the irradiance at the solar surface. $\frac{\text{Sun\_Radius}^2}{\text{Astronomical\_Unit}^2}$



Name	F
Function	fb0003
Inputs	0
Outputs	1
Parameters	0
Strings	0
Group	C

Inputs

None

Outputs

1 Dilution factor f

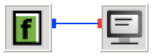
Parameters

None

Strings

None

f.vseit

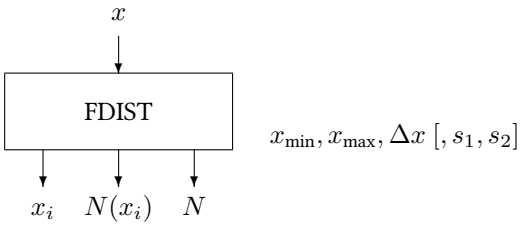


The F block provides the dilution factor which is displayed by the SCREEN block.

```
2.16448861E-05
```

1.54 Block FDIST

The FDIST block calculates the frequency distribution of its input signal.



Name	FDIST
Function	fb0043
Inputs	1
Outputs	4
Parameters	3 . . . [5]
Strings	0
Group	T

Inputs

1 Any signal x

Outputs

1 Centre of a bin x_i
2 Normalised number of data in corresponding bin $N(x_i)$
3 Total number of data N
4 Absolute number of data in corresponding bin $N(x_i)$

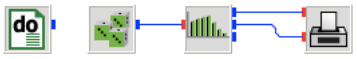
Parameters

1 Minimum $x_{\min}$ of distribution interval
2 Maximum $x_{\max}$ of distribution interval
3 Width Δx of bins; the total number of bins may not exceed 1000
4 Switch s_1 to suppress output of data outside the interval $[x_{\min}, x_{\max}]$. If s is set to 1 output is suppressed, otherwise the data are given for $x_{\min} - \Delta x/2$ or $x_{\max} + \Delta x/2$ by default, respectively
5 Switch s_2 to define the bin intervals right open [) by setting s_2 to 0 (default) or left open (] by setting s_2 to 1

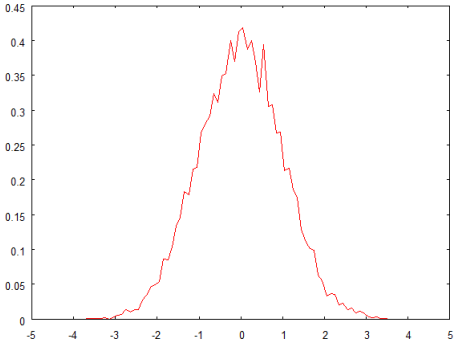
Strings

None

fdist.vseit

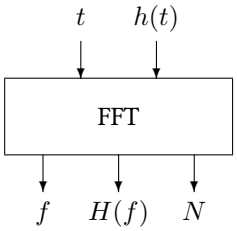


A DO block delivers 1000 clicks, i. e., counts from 1 to 1000. The GASDEV block generates one normally distributed random number for each step. At the end of this process the FDIST block calculates the frequency distribution and the PLOT block displays the graph on screen.



1.55 Block FFT

The FFT block calculates the discrete Fourier transform of its input signal.



Alogrithm

Name	FFT
Function	fb0053
Inputs	2
Outputs	3
Parameters	0 . . . [1]
Strings	0
Group	T

Inputs

1 Any signal t
2 Any signal $h(t)$

Outputs

1 Inverse of t
2 Fourier transform $H(f)$
3 Total number of data points

Parameters

1 FFT algorithm
0 Cooley-Tuckey algorithm

Strings

None

Fourier transform

A physical process can be described either in the time domain $h(t)$ or else in the fregquency domain $H(t)$. The equations for the transformations are given by

$$H(f) = \int_{-\infty}^{\infty} h(t) \exp(2\pi i f t) \mathrm{d}t$$

and

$$h(t) = \int_{-\infty}^{\infty} H(f) \exp(-2\pi i f t) \mathrm{d}f$$

respectively. If t is measured in seconds, the unit of f is in s^{-1} or Hertz. The unit of t is not restricted to time. If t is given in meters, for example, the unit of f is m^{-1}.

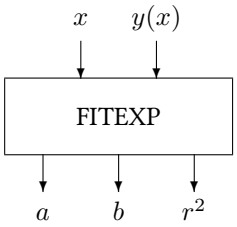
Remark

The use of the FFT block is restricted to a maximum of $2^{20} = 1\,048\,576$ data points in the current version.

The block is still under construction.

1.56 Block FITEXP

The FITEXP block approximates the input data by an exponential function.



Name	FITEXP
Function	fb0034
Inputs	2
Outputs	3
Parameters	0
Strings	0
Group	I

Inputs

1 Any value x
2 Corresponding $y(x)$ value

Outputs

1 Fit coefficient a
2 Fit coefficient b
3 Regression coefficient r^2

Parameters

None

Strings

None

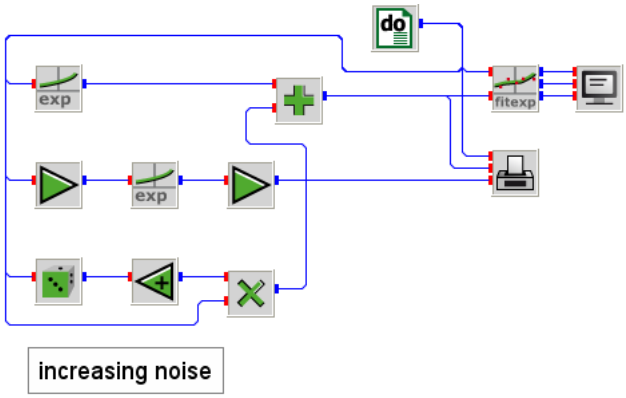
Description

The blocks input data are approximated by the exponential function

$$y = a \exp(bx)$$

The successors of this block are executed only at the end of the simulation run.

fitexp.vseit



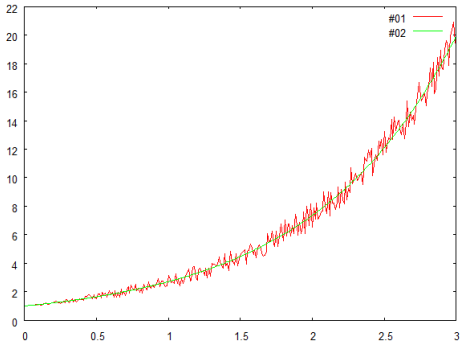
A DO block runs x values from 0 to 3 in steps of 0.01. The data are used by an EXP block to calculate $\exp(x)$ to which some noise is generated in the lower part of the block diagram and added to $\exp(x)$. This signal is used by the FITEXP block to find $a = 1$ and $b = 1$, theoretically.

The noise is generated by a RAN1 block with an offset of -0.5 calculated by the OFFSET block. A MUL block multiplies the noise by the output of the DO block to make it increasing in amplitude as x grows.

The outputs a, b, and r^2 are displayed numerically by a SCREEN block.

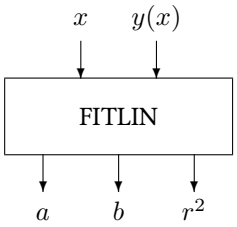
```
0.99717307      0.99807161      0.99165821
```

Assuming that the result is already known ($a = 0.99717307$ and $b = 0.99807161$) the exponential equation $y' = a\exp(bx)$ is calculated and plotted against the input data by a PLOT block.



1.57 Block FITLIN

The FITLIN block approximates the input data by a linear function.



Name	FITLIN
Function	fb0033
Inputs	2
Outputs	3
Parameters	0
Strings	0
Group	I

Inputs

1 Any value x
2 Corresponding $y(x)$ value

Outputs

1 Fit coefficient a
2 Fit coefficient b
3 Regression coefficient r^2

Parameters

None

Strings

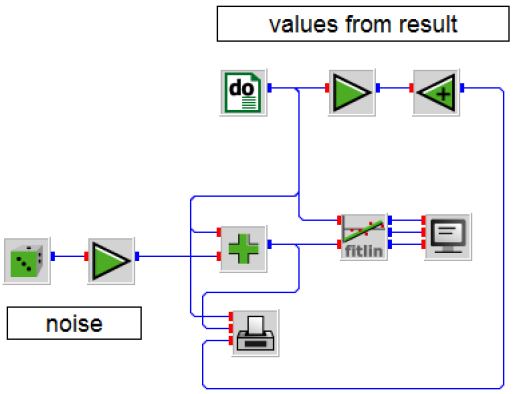
None

Description The blocks input data are approximated by the linear equation

$$y = a + bx$$

The successors of this block are executed only at the end of the simulation run.

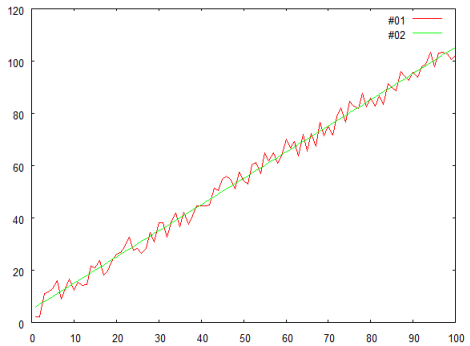
fitlin.vseit



A DO block delivers 100 clicks, i. e., counts from 1 to 100. A RAN1 block generates one uniformly distributed random number for each step multiplied by 10 by a GAIN block. A SUM block adds the DO block's output and the output of the GAIN block giving the variable to be fitted as a function of the DO block's output by the FITLIN block. The outputs a, b, and r^2 are then displayed numerically by a SCREEN block.

```
4.8319  1.0039  0.9903
```

Assuming that the result is already known ($a = 4.8319$ and $b = 1.0039$) the linear equation $y' = a + bx$ is calculated. A PLOT block displays the data and the fit on screen.



Remarks The FITLIN block is a typical example for an I-block: Its successors are executed (once) only when a certain condition is fullfilled—here the end of the main simulation loop. Compare this behavior with a typical example for a T-block like block FDIST, for example.

1.58 Block FITLLS

The FITLLS block approximates the input data by a linear least quare fit.

Name	FITLLS
Function	fb0077
Inputs	2 *ldots* [11]
Outputs	1 *ldots* [10]
Parameters	0
Strings	0
Group	I

Inputs

1 Any value y_1
2 Any value x_1
3 Any value x_2
max+1 Any value $x_{\max}$

Outputs

1 Fit coefficient a_1
2 Fit coefficient a_2
max Fit coefficient $a_{\max}$
max+1 Regression coefficient r^2

Parameters

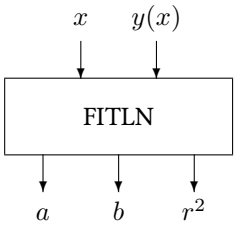
None

Strings

None

1.59 Block FITLN

The FITLN block approximates the input data by a logarithmic function.



Name	FITLN
Function	fb0035
Inputs	2
Outputs	3
Parameters	0
Strings	0
Group	I

Inputs

1 Any value x
2 Corresponding $y(x)$ value

Outputs

1 Fit coefficient a
2 Fit coefficient b
3 Regression coefficient r^2

Parameters

None

Strings

None

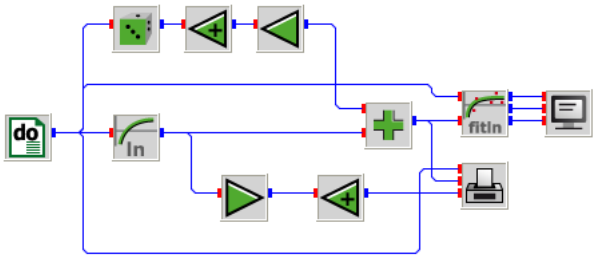
Description

The blocks input data are approximated by the logarithmic function

$$y = a + b \ln x$$

The successors of this block are executed only at the end of the simulation run.

fitln.vseit



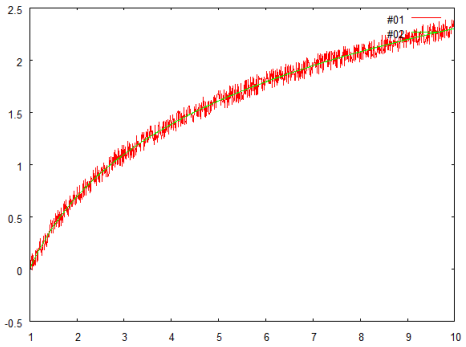
A DO block runs x values from 1 to 10 in steps of 0.01. The data are used by an LN block to calculate $\ln x$ to which some noise is generated in the upper part of the block diagram and added to $\ln x$. This signal is used by the FITLN block to find $a = 0$ and $b = 1$, theoretically.

The noise is generated by a RAN1 block with an offset of -0.5 calculated by the OFFSET block. An attenuator block ATT divides the noise by 5.

The outputs a, b, and r^2 are displayed numerically by a SCREEN block.

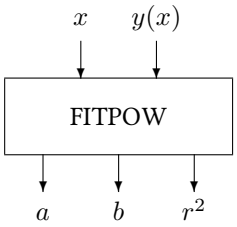
```
-0.000548    0.999736    0.990468
```

Assuming that the result is already known ($a = -0.000548$ and $b = 0.999736$) the logarithmic equation $y' = a + b \ln x$ is calculated and plotted against the input data by a PLOT block.



1.60 Block FITPOW

The FITPOW block approximates the input data by a power function.



Name	FITPOW
Function	fb0036
Inputs	2
Outputs	3
Parameters	0
Strings	0
Group	I

Inputs

1 Any value x
2 Corresponding $y(x)$ value

Outputs

1 Fit coefficient a
2 Fit coefficient b
3 Regression coefficient r^2

Parameters

None

Strings

None

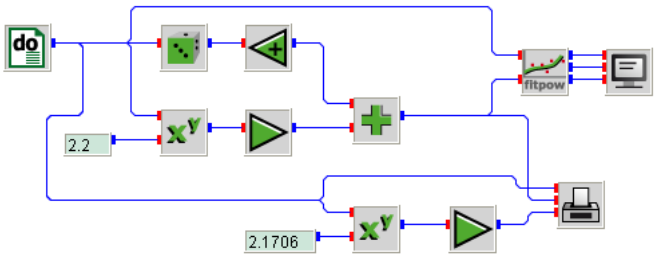
Description

The blocks input data are approximated by the power function

$$y = ax^b$$

The successors of this block are executed only at the end of the simulation run.

fitpow.vseit



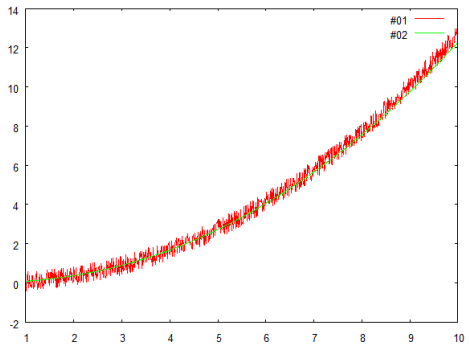
A DO block runs x values from 1 to 10 in steps of 0.01. The data are used by an EXPG block with exponent 2.2 as defined by a connected CONST block. The result is multiplied by a factor 0.08 by a GAIN block. Some noise is generated in the upper part of the block diagram and added to $0.08\,x^{2.2}$. This signal is used by the FITPOW block.

The noise is generated by a RAN1 block with an offset of -0.5 calculated by the OFFSET block.

The outputs a, b, and r^2 are displayed numerically by a SCREEN block.

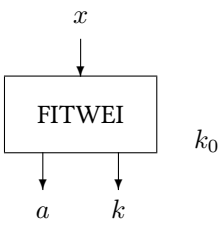
```
8.29010606E-02   2.1706350      0.88110483
```

Assuming that the result is already known ($a = 0.0829$ and $b = 2.1706$) the logarithmic equation $y' = ax^b$ is calculated and plotted against the input data by a PLOT block.



1.61 Block FITWEI

The FITWEI block approximates the input data by a Weibull distribution.



Name	FITWEI
Function	fb0059
Inputs	1
Outputs	2
Parameters	0
Strings	0
Group	I

Inputs

1 Any value x

Outputs

1 Scale parameter a
2 Shape parameter k

Parameters

None

Strings

None

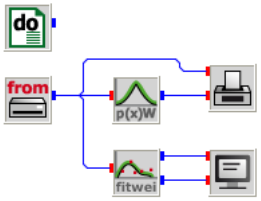
Description

The blocks input data are approximated by the Weibull distribution.

$$p(x) = \frac{k}{a}\left(\frac{x}{a}\right)^{k-1} \exp\left(-\frac{x^k}{a^k}\right)$$

where a denotes the scale parameter and k the shape parameter. The successors of this block are executed only at the end of the simulation run.

fitwei.vseit

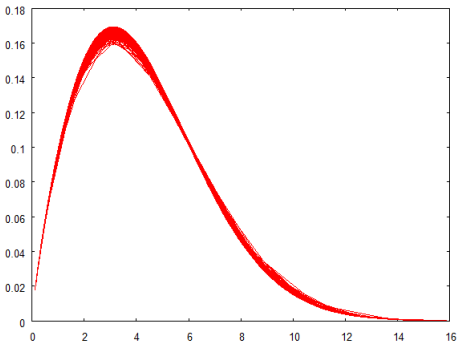


A DO block runs through the 8760 hours of a year to trigger a READ block which reads all records of the file `vwBerlin.dat`, saved in the `examples/data` directory. This file contains wind speed data which have been generated by the GENV block.

The FITWEI block determines the shape and scale parameters of the Weibull distribution, which are displayed by a SCREEN block.

```
4.80069113  1.83100331
```

Assuming that the result is already known ($a = 4.8007$ and $k = 1.831$) the Weibull distribution is calculated by a DISWEI block and plotted by a PLOT block.

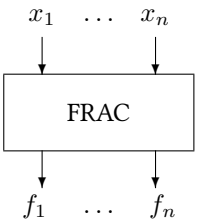


Remark Please observe that the “thick” shape of the graph as displayed by the PLOT block is due to the fact, that all data points are connected by a straight line by default.

1.62 Block FRAC

The FRAC block returns the decimal fraction of the input signal.

12.34



Name	FRAC
Function	fb0005
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	0
Strings	0
Group	S

Inputs

i Any value x_i

Outputs

i Decimal fraction f of x_i, i. e., f is set to $x_i - \text{int}(x_i)$.

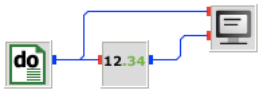
Parameters

None

Strings

None

frac.vseit

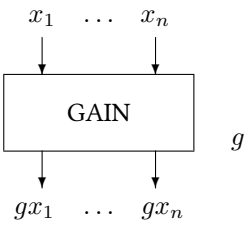


A DO block provides numbers from zero to two in steps of 0.04. The FRAC block calculates the decimal fraction which is then displayed by SCREEN block.

```
0.00    0.00
0.40    0.40
0.80    0.80
1.20    0.20
1.60    0.60
2.00    0.00
```

1.63 Block GAIN

The GAIN block multiplies its input by its parameter.



Name	GAIN
Function	fb0006
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	1
Strings	0
Group	S

Inputs

i Any value x_i

Outputs

i Product of input and parameter gx_i

Parameters

1 Gain factor g

Strings

None

gain.vseit

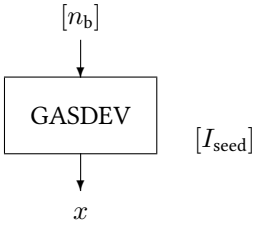
The constant two provided by the CONST block is multiplied by the factor 3.14 of the GAIN block whose output is then displayed by the SCREEN block.

```
6.28
```

See also Block ATT.

1.64 Block GASDEV

The GASDEV block returns a normally distributed deviate with zero mean and unit variance, using RAN1 as the source of uniform deviates.



Name	GASDEV
Function	fb0047
Inputs	0 . . . [1]
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Block number n_b of a predecessor (optional)

Outputs

1 Normally distributed deviate x with zero mean and unit variance

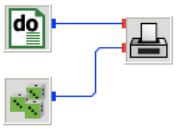
Parameters

1 Initialization I_{seed} of random number generator, set to 1536 by default; the sign of I_{seed} is ignored

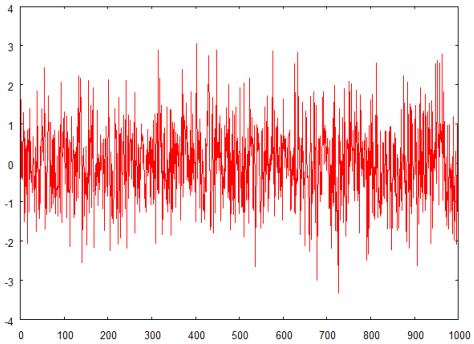
Strings

None

gasdev.vseit



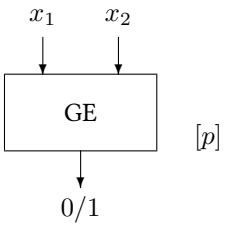
In a DO block 1000 clicks are generated by the parameter setting. For each click the GASDEV block generates one normal distributed random number. The output of the DO block and the output of the GASDEV block is then plotted by the PLOT block.



See also Block RAN1.

1.65 Block GE

The GE block tests wheter its first input is greater than or equal to the second one.



Name	GE
Function	fb0027
Inputs	2
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Any value x_1
2 Any value x_2

Outputs

1 y is set to one if x_1 is greater than x_2 or $|x_1 - x_2| \leq p$, otherwise y is set to zero

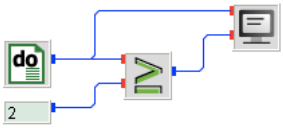
Parameters

1 Error tolerance $p \geq 0$ for the difference between x_1 and x_2, set to zero by default

Strings

None

ge.vseit

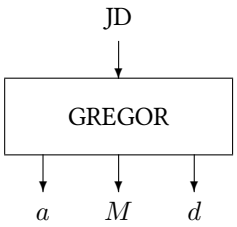


A DO block counts from 1 to 3. The CONST block provides a constant with value 2. The GE blocks checks, whether the output of the DO block is greater or equal to 2 (true = 1) or not (false = 0). The SCREEN block displays the result.

```
1.0   0.0
2.0   1.0
3.0   1.0
```

1.66 Block GREGOR

The GREGOR block returns the Gregorian Date for a given Julian Day.



Name	GREGOR
Function	fb0026
Inputs	1
Outputs	3
Parameters	0
Strings	0
Group	S

Inputs

1 Julian Day JD

Outputs

1 Year a
2 Month M
3 Day d

Parameters

None

Strings

None

gregor.vseit

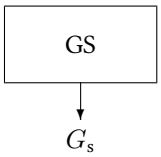
The example shows that the Julian Day 2455198, defined by a CONST block, corresponds to the First of January 2010.

```
2010.    1.    1.
```

See also Block JULIAN.

1.67 Block GS

The GS block provides the solar constant.



Name	GS
Function	fb0003
Inputs	0
Outputs	1
Parameters	0
Strings	0
Group	C

Inputs

None

Outputs

1 Solar constant G_s / $\mathrm{W\,m^{-2}}$.

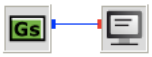
Parameters

None

Strings

None

gs.vseit

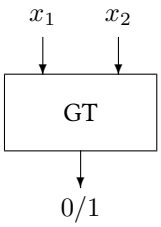


The GS block provides the solar constant which is displayed by the SCREEN block.

```
1367.0 W/m^2
```

1.68 Block GT

The GT block tests whether its first input is greater than the second one.



Name	GT
Function	fb0028
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any value x_1
2 Any value x_2

Outputs

1 y is set to one if x_1 is greater than x_2, otherwise y is set to zero

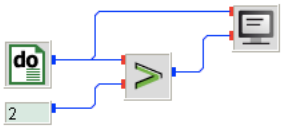
Parameters

None

Strings

None

gt.vseit

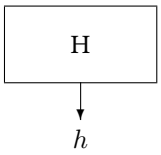


A DO block counts from 1 to 3. The CONST block provides a constant with value 2. The GT blocks checks, whether the output of the DO block is greater than 2 (true = 1) or not (false = 0). The SCREEN block displays the result.

```
1.0   0.0
2.0   0.0
3.0   1.0
```

1.69 Block H

The H block provides the Planck constant.



Name	H
Function	fb0003
Inputs	0
Outputs	1
Parameters	0
Strings	0
Group	C

Inputs

None

Outputs

1 Planck constant h / J s

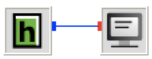
Parameters

None

Strings

None

h.vseit

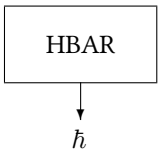


The H block provides the Planck constant which is displayed by the SCREEN block.

```
6.62617579E-34
```

1.70 Block HBAR

The HBAR block provides the reduced Planck constant.



Name	HBAR
Function	fb0003
Inputs	0
Outputs	1
Parameters	0
Strings	0
Group	C

Inputs

None

Outputs

1	Reduced Planck constant $\hbar = \frac{h}{2\pi}$ / J s

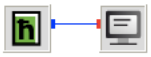
Parameters

None

Strings

None

hbar.vseit

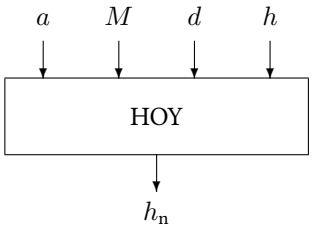


The HBAR block provides the reduced Planck constant which is displayed by the SCREEN block.

```
1.05458856E-34
```

1.71 Block HOY

The HOY block returns the hour of the year.



Name	HOY
Function	fb0023
Inputs	4
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Year a
2	Month M
3	Day d
4	Hour h

Outputs

1	Hour of the year $h_n \in [1, 8784]$

Parameters

None

Strings

None

hoy.vseit

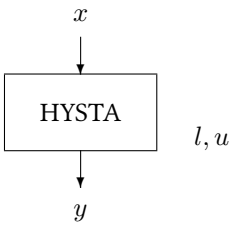
A CLOCK block provides the last hour of 31 December 2012. The HOY block calculates the hour of the year. The hour of the year is then displayed by the SCREEN block.

```
A leap year has 8784. hours
```

See also Blocks DOY, MOY. and SOY.

1.72 Block HYSTA

The HYSTA block simulates an anti-clockwise hysteris controller.



Name	HYSTA
Function	fb0071
Inputs	1
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Any signal x

Outputs

1 Hysterestis signal y

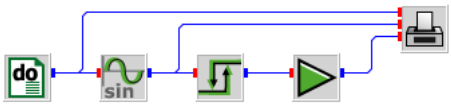
Parameters

1 Lower threshold l
2 Upper threshold u

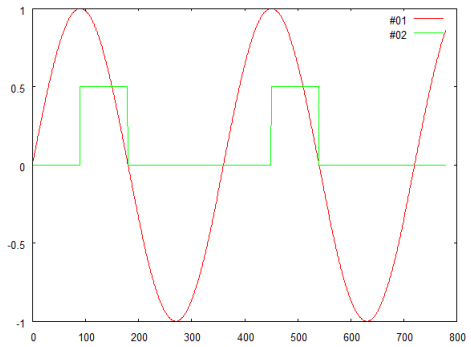
Strings

None

hysta.vseit



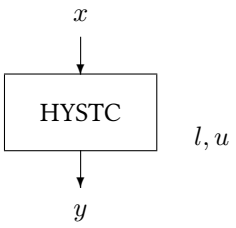
A DO block runs a SIN block with values for an angle α from 0 to 780 degrees in steps of 1. The $\sin(\alpha)$ data are used by a HYSTA block with lower threshold zero and upper threshold 1. The result is multiplied by a factor 0.5 by a GAIN block. The $\sin(\alpha)$ data together with the output of the GAIN block are plotted by a PLOT block.



It can be seen, that the output HYSTA block remains on the lower level zero until the upper threshold one is reached. As expected, the output remains on the upper level until the lower threshold is reached. The “switch” remains “off” until the upper threshold one is reached again. This is the typical behavior of a on/off switch with hysteris.

1.73 Block HYSTC

The HYSTC block simulates a clockwise hysteris controller.



Name	HYSTC
Function	fb0071
Inputs	1
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Any signal x

Outputs

1 Hysterestis signal y

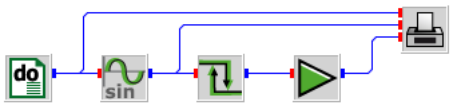
Parameters

1 Lower threshold l
2 Upper threshold u

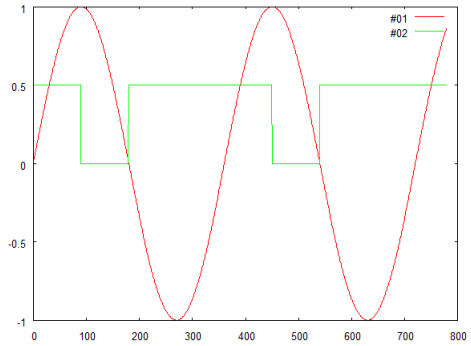
Strings

None

hystc.vseit



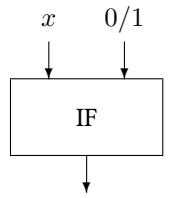
A DO block runs a SIN block with values for an angle α from 0 to 780 degrees in steps of 1. The $\sin(\alpha)$ data are used by a HYSTC block with lower threshold zero and upper threshold 1. The result is multiplied by a factor 0.5 by a GAIN block. The $\sin(\alpha)$ data together with the output of the GAIN block are plotted by a PLOT block.



It can be seen, that the output HYSTC block remains on the upper level one until the upper threshold one is reached. As expected, the output remains on the lower level until the lower threshold is reached. The “switch” remains “on” until the upper threshold one is reached again. This is the typical behavior of a on/off switch with hysteris.

1.74 Block IF

The IF block returns its first input if the second input is equal to one (true), and does not return anything otherwise. In contrast, IFELSENAN returns *NaN* if the second input if zero. WARNING: The successors of this block are completely ignored when the second signal is false. This behaviour is different than in many other programming languages. An alternative is to use a MUL block instead of an IF block.



Name	IF
Function	fb0022
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	I

Inputs

1 Any signal x
2 Logical input either 0 (false) or $\neq 0$ (true).

Outputs

1 Value of x. The successors of this block are executed only if the second input is not equal to 0.

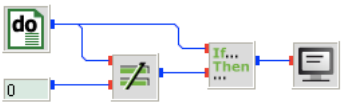
Parameters

None

Strings

None

if.vseit



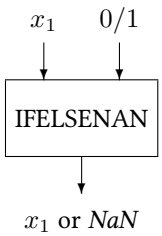
In a DO block numbers are counted from −1 to 1. A CONST returns a constant value 0. The NE block compares the output of the DO block with the constant. If they are different the IF block lets the output of the DO block pass and the SCREEN block displays the value, otherwise the output is suppressed because the execution of the SCREEN block is skipped.

```
-1.0
 1.0
```

1.75 Block IFELSENAN

IFELSENAN expects a signal x_1, and one logical signal x_2. If x_2 is true, x_1 is returned. If x_2 is false, *NaN* is returned.

This behaviour can be useful in order to indicate missing data, e.g. for plots. IF block is similar, but simply doesn't return anything if x_2 is false.



Name	IFELSENAN
Function	fb0085
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Any signal x_1
2	Any logical signal x_2

Outputs

1	x_1 or *NaN*

Parameters

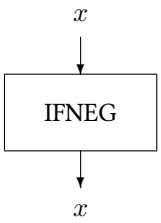
None

Strings

None

1.76 Block IFNEG

The IFNEG block returns its input signal only if it is negative.



Name	IFNEG
Function	fb0046
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	I

Inputs

1 Any signal x

Outputs

1 Signal x in case of $x < 0$, otherwise the successors of this block are not executed.

Parameters

None

Strings

None

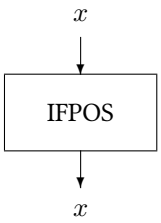
ifneg.vseit

In a DO block numbers are counted from -1 to $+1$. The IFNEG block checks whether the output of the DO block is negative. If yes, the SCREEN block displays the value, otherwise execution of the SCREEN block is skipped.

```
-1.0
```

1.77 Block IFPOS

The IFPOS block returns its input signal only if it is positive.



Name	IFPOS
Function	fb0046
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	I

Inputs

1 Any signal x

Outputs

1 Signal x in case of $x > 0$, otherwise the successors of this block are not executed.

Parameters

None

Strings

None

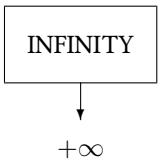
ifpos.vseit

In a DO block numbers are counted from -1 to $+1$. The IFPOS block checks whether the output of the DO block is positive. If yes, the SCREEN block displays the value, otherwise execution of the SCREEN block is skipped.

```
1.0
```

1.78 Block INFINITY

The INFINITY block returns +Infinity, which could be useful for testing purposes.



Name	INFINITY
Function	fb0003
Inputs	0
Outputs	1
Parameters	0
Strings	0
Group	C

Inputs

None

Outputs

1 +∞.

Parameters

None

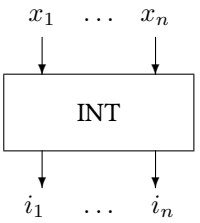
Strings

None

1.79 Block INT

The INT block returns the integer value of the input signal.

12.34



Name	INT
Function	fb0005
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	0
Strings	0
Group	S

Inputs

i Any value x_i

Outputs

i Truncated integer value of x_i, ie if $n \leq x_i < n + 1$ for any value $n \in \mathbf{N}$, return value is n.

Parameters

None

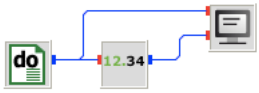
Strings

None

Remarks

The output of the INT block has a rounded value but is still a Fortran Real variable.

int.vseit

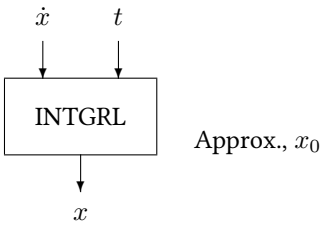


A DO block varies its output from 0 to 2 in steps of 0.4. The INT block calculates the integer fraction which is then displayed by a SCREEN block.

```
0.00    0.00
0.40    0.00
0.80    0.00
1.20    1.00
1.60    1.00
2.00    2.00
```

1.80 Block INTGRL

The INTGRL block calculates the integral of its input signal.



Name	INTGRL
Function	fb0065
Inputs	2
Outputs	1
Parameters	1
Strings	0
Group	D

Inputs

1 $\dot{x}$ Derivative of a signal x
2 Time t

Outputs

1 Integral approximation $x \approx \int_t^{t+\Delta t} x(t)\mathrm{d}t$

Parameters

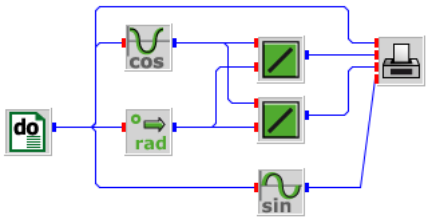
1 Integration method
- 0 Euler
- 1 Heun

2 Initial value x_0

Strings

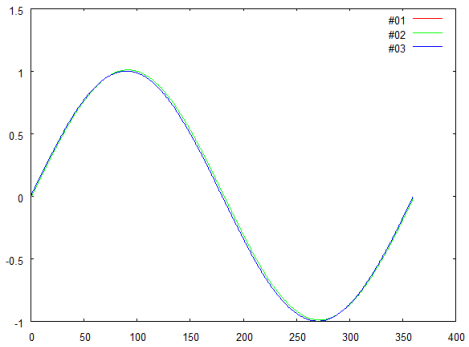
None

intgrl.vseit



A DO block varies an angle α from 0 to 360 degrees in steps of one degree and delivers it to the connected COS and DEG2RAD blocks. Both $\cos(\alpha)$ and α in radians are connected to the inputs of two INTGRL blocks with initial value zero, the upper one uses Euler's integration method, the lower one Heun's.

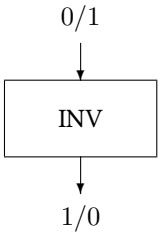
Since we know that $\int \cos(\alpha)\, d\alpha = \sin(\alpha)$ a SIN block is used to compare the theoretical result with the two approximations. The result is shown by a PLOT block.



The curves can hardly be distinguished.

1.81 Block INV

The INV block inverts the input signal logically.



Name	INV
Function	fb0031
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any value x close to one or zero

Outputs

1 y is set to zero if $x \in [0.5, 1.5)$ otherwise y is set to one

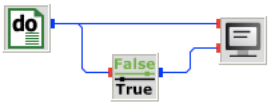
Parameters

None

Strings

None

inv.vseit

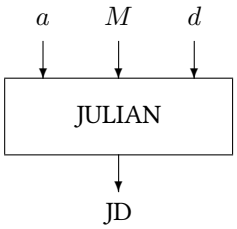


In a DO block numbers are counted from 0 to 2. The INV block inverts the result logically and the SCREEN block displays the results.

```
   0.0   1.0
   1.0   0.0
W05052 Block 00001: Invalid non logical input
   2.0   1.0
W05053 Block 00001: Calls with invalid non logical input:        1
```

1.82 Block JULIAN

The JULIAN block returns the Julian Day for a given Gregorian Date.



Name	JULIAN
Function	fb0025
Inputs	3
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Year a
2	Month M
3	Day d

Outputs

1	Julian Day JD

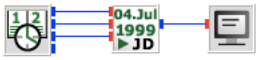
Parameters

None

Strings

None

julian.vseit



The example shows that the First of January 2010 defined by a CLOCK block corresponds to the Julian Day

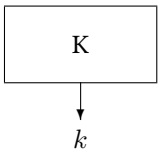
```
2455198.
```

as shown by the SCREEN block.

See also Block GREGOR.

1.83 Block K

The K block provides the Boltzmann constant.



Name	K
Function	fb0003
Inputs	0
Outputs	1
Parameters	0
Strings	0
Group	C

Inputs

None

Outputs

1 Boltzmann constant k / $\mathrm{J\,K^{-1}}$

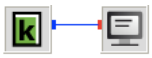
Parameters

None

Strings

None

k.vseit

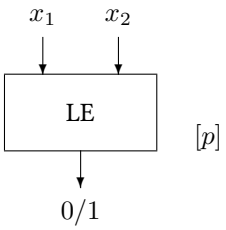


The K block provides the Boltzmann constant which is displayed by the SCREEN block.

```
1.38065994E-23
```

1.84 Block LE

The LE block tests whether its first input signal is less than or equal to the second one.



Name	LE
Function	fb0027
Inputs	2
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Any value x_1
2 Any value x_2

Outputs

1 y is set to one if x_1 is less than x_2 or $|x_1 - x_2| \leq p$, otherwise y is set to zero

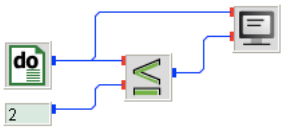
Parameters

1 Error tolerance $p \geq 0$ for the difference between x_1 and x_2, set to zero by default

Strings

None

le.vseit

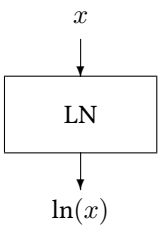


A DO block counts from 1 to 3. The CONST block provides a constant with value 2. The LE block checks, whether the output of the DO block is less or equal to 2 (true = 1) or not (false = 0). The SCREEN block displays the result.

```
1.0   1.0
2.0   1.0
3.0   0.0
```

1.85 Block LN

The LN block returns the natural (base e) logarithm of its input.



Name	LN
Function	fb0011
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any value $x > 0$

Outputs

1 Natural logarithm $\ln(x)$

Parameters

None

Strings

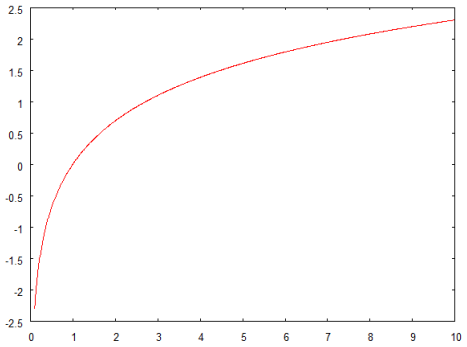
None

Remarks

If the input value x is zero or negative, the LN block displays a warning message and performs no operation. If this happens more than once, the warning message is suppressed. At the end of a simulation run the total number of occurencies is displayed.

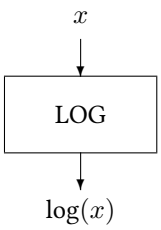
ln.vseit

A DO block varies x from 0.1 to 10 in steps of 0.01. The LN block calculates the natural logarithm (base e) and the PLOT block displays the graph on screen.



1.86 Block LOG

The LOG block returns the common (base 10) logarithm of its input.



Name	LOG
Function	fb0011
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any value $x > 0$

Outputs

1 Common logarithm $\log(x)$

Parameters

None

Strings

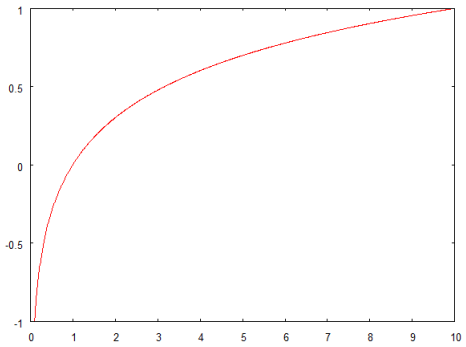
None

Remarks

If the input value x is zero or negative, the LOG block displays a warning message and performs no operation. If this happens more than once, the warning message is suppressed. At the end of a simulation run the total number of occurencies is displayed.

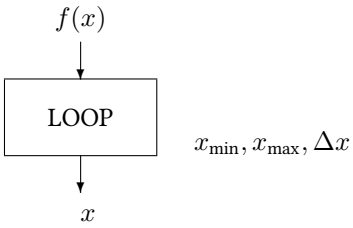
log.vseit

A DO block varies x from 0.1 to 10 in steps of 0.01. The LOG block calculates the logarithm (base 10) and the PLOT block displays the graph on screen.



1.87 Block LOOP

The LOOP block runs through a sequence of values defined by parameters, restricted to a part of the simulation model.



Name	LOOP
Function	fb0055
Inputs	1
Outputs	1
Parameters	3
Strings	0
Group	L

Inputs

1 Any signal $y = f(x)$ within the loop

Outputs

1 Signal x which is varied in steps as defined through the parameters

Parameters

1 Minimum output signal $x_{\min}$
2 Maximum output signal $x_{\max}$
3 Increment of x per step Δx

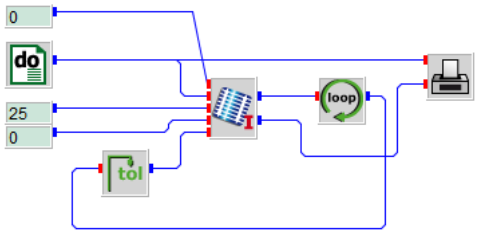
Strings

None

Description

The output signal of the LOOP block runs from $x_{\min}$ to $x_{\max}$ in steps of Δx. The function is very similar to that of the DO block with one difference: DO is a T-block, while LOOP is an L-block, which can set up a local loop.

loop.vseit This example shows how a LOOP block can be used in the calculation of steady-state PV module temperatures as a function of the global radiation.

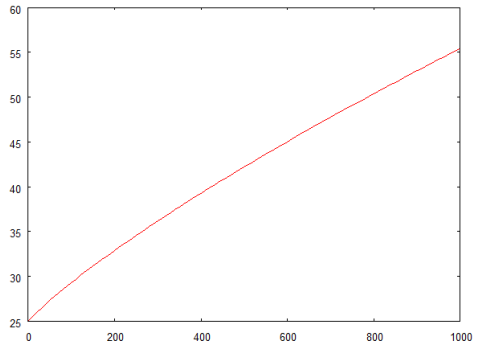


The global radiation is varied from 0 to 1000 Watt per square meter in steps of 10 by the DO block. The output of the DO block is used by a PVI block which calculates the current and module temperature in DEQ temperature mode. Ambient temperature and wind speed are set constant by CONST blocks.

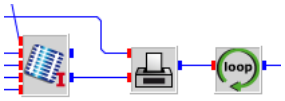
The LOOP block sets up a local loop. Since its output is used as time input of the PVI block the parameters are set to a start of 0, stop of 3600 and increment 60 seconds, i. e., the LOOP block simulates one hour during which the PV module can heat up. Since the voltage is set to zero the PVI block simulates short-circuit-current operation.

Please notice that the feedback connection between LOOP and PVI block does not set up an algebraic loop. This is a feature of L-blocks in combination with a TOL block.

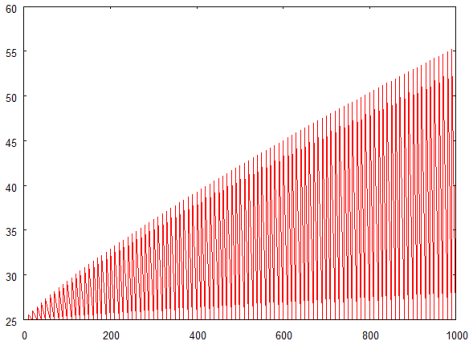
The PLOT block shows the final result.



The x coordinate is the solar radiation in Watt per square meter, the y coordinate the steady-state module temperature in degrees Celsius.



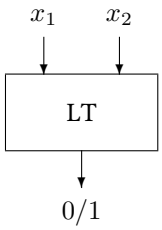
If the PLOT block is placed between the PVI block and the LOOP block the iteration process for each radiation value can be observed.



See also Block DO, block MPP, and block NULL, for instance.

1.88 Block LT

The LT block tests whether its first input is less than the second one.



Name	LT
Function	fb0028
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any value x_1
2 Any value x_2

Outputs

1 y is set to one if x_1 is less than x_2, otherwise y is set to zero

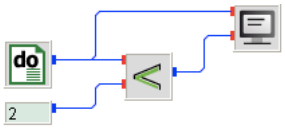
Parameters

None

Strings

None

lt.vseit



A DO block counts from 1 to 3. The CONST block provides a constant with value 2. The LT blocks checks, whether the output of the DO block is less than 2 (true = 1) or not (false = 0). The SCREEN block displays the result.

```
1.0   1.0
2.0   0.0
3.0   0.0
```

1.89 Block MAE

The MAE block calculates the mean absolute error between the input signals over a complete simulation run.

Name	MAE
Function	fb0075
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	I

Inputs

1 Simulated time series x_s
2 Measured time series x_m

Outputs

1 Mean absolue error

Parameters

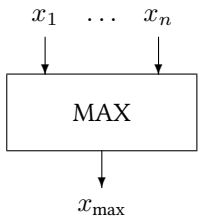
None

Strings

None

1.90 Block MAX

The MAX block returns the maximum of its inputs.



Name	MAX
Function	fb0002
Inputs	1 . . . [999]
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Any value x_1
2	Any value x_2
n	Any value x_n

Outputs

1	Maximum $x_{\max}$ of all x_i for $i = 1 \ldots n \leq 999$

Parameters

None

Strings

None

Description The maximum $x_{\max}$ is calculated from the inputs $x_1 \ldots x_n$ according to

$$x_{\max} = \max\{x_i | i = 1 \ldots n\}$$

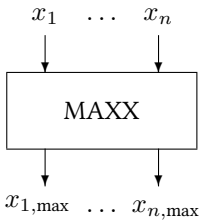
max.vseit

Two CONST blocks provide the constants 1 and 3.14. The MAX blocks checks, which of the inputs is greater. The SCREEN block displays the result.

```
3.14
```

1.91 Block MAXX

The MAXX block finds its maximum input signal over a complete simulation run.



Name	MAXX
Function	fb0021
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	0
Strings	0
Group	I

Inputs

i Any signal x_i

Outputs

i Maximum over all x_i

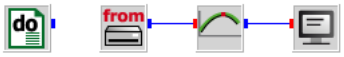
Parameters

None

Strings

None

maxx.vseit



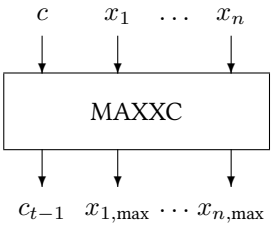
A DO block provides 8760 steps to read ambient temperature data by a READ block from file `temperature.dat` located in the `data` directory.

The MAXX block detects the maximum value and provides it to a SCREEN block to show the result.

32.2

1.92 Block MAXXC

The MAXXC block scans its second input signal for a maximum as long as the condition input remains constant. When this input changes its value, the maximum is on output and the successors of the block are executed.



Name	MAXXC
Function	fb0037
Inputs	2 . . . [51]
Outputs	2 . . . [51]
Parameters	0
Strings	0
Group	I

Inputs

1	Condition input c
2	Any signal x_1
n	Any signal x_{n-1}

Outputs

1	Condition input c delayed by one step
2	Maximum $\max_1$ over all $c =$ const calls
n	Maximum $\max_{n-1}$ over all $c =$ const calls

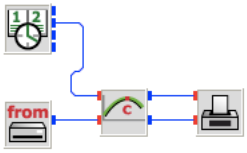
Parameters

None

Strings

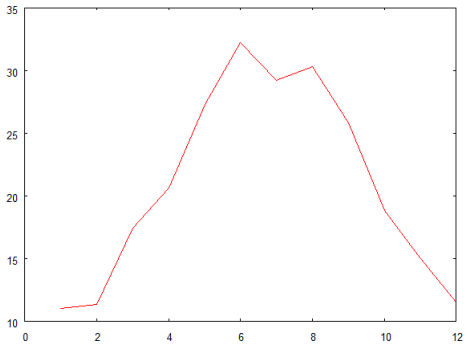
None

maxxc.vseit



A CLOCK block runs through one year in steps of one hour and controls reading the data file `meteo82.dat` from the `examples/data` directory as specified by the parameters of a READ block. By definition of the Format parameter of the READ block `(33X,F5.1,26X)` only colums 34–38 are evaluated. These contain ambient temperature data in degrees Celsius.

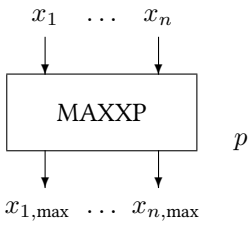
Since the month output of the CLOCK block is used as condition input to the MAXXC block, monthly maxima of the temperature data are calculated by the MAXXC block and finally displayed as graph by the PLOT block.



See also Block MINNC.

1.93 Block MAXXP

The MAXXP block finds the maximum of its input signal over a number of steps as defined through its parameter.



Name	MAXXP
Function	fb0020
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	1
Strings	0
Group	I

Inputs

i Any signal x_i

Outputs

i Maximum $x_{i,\max}$ of x_i

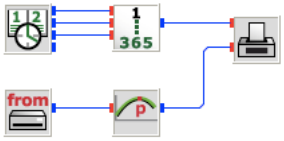
Parameters

1 Number p of steps to find the maximum $x_{i,\max}$

Strings

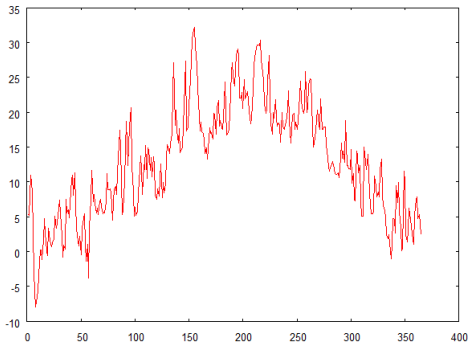
None

maxxp.vseit



A CLOCK block runs through one year in steps of one hour and controls reading the data file `meteo82.dat` from the `examples/data` directory as specified by the parameters of a READ block. By definition of the Format parameter of the READ block `(33X,F5.1,26X)` only colums 34–38 are evaluated. These contain ambient temperature data in degrees Celsius.

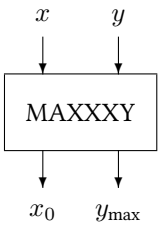
Since the parameter of the MAXXP block is set to 24, daily maxima of the temperature data are calculated by the MAXXP block and finally displayed as graph by the PLOT block.



See also Block MINNP.

1.94 Block MAXXXY

The MAXXXY block finds the maximum of its second input over a complete simulation run and returns the x and y coordinates.



Name	MAXXXY
Function	fb0070
Inputs	2
Outputs	2
Parameters	0
Strings	0
Group	I

Inputs

1 Any signal x
2 Corresponding value $y(x)$

Outputs

1 x_0 coordinate of maximum
2 Maximum value $y_{max} = y(x_0)$

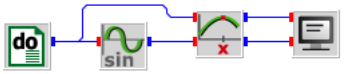
Parameters

None

Strings

None

maxxxy.vseit



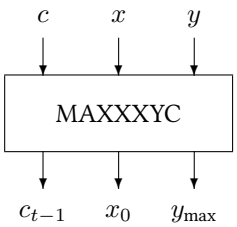
A DO block provides valaues for an angle α between 0 and 180 degrees in steps of one, which are used by a SIN block to calculate $\sin(\alpha)$.

The MAXXXY block detects the maximum value and outpus the coordinate of the maximum and the value of the maximum to a SCREEN block to show the result.

```
89.98999786  1.00000000
```

1.95 Block MAXXXYC

The MAXXXYC block scans its third input signal for a maximum as long as the condition input remains constant. When this input changes its value, the maximum and the corresponding coordinate is on output and the successors of the block are executed.



Name	MAXXXYC
Function	fb0073
Inputs	3
Outputs	3
Parameters	0
Strings	0
Group	I

Inputs

1 Condition input c
2 Any signal x
3 Corresponding signal y

Outputs

1 Condition input c delayed by one step
2 x-oordinate of maximum value
3 Maximum over all $c =$ const calls

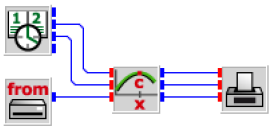
Parameters

None

Strings

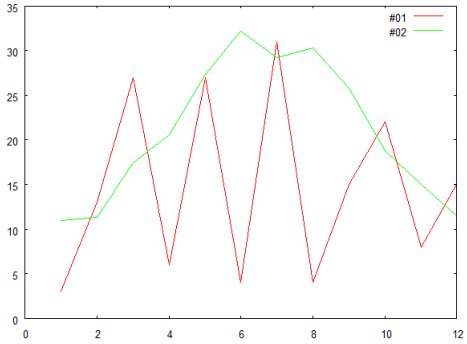
None

maxxxyc.vseit



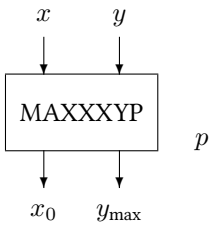
A CLOCK block runs through the hours of the year 1982. Corresponding temperature data are read by a READ block from file `meteo82.dat` which can be found in the `examples/data` directory.

The MAXXXYC block detects the monthly maximum values and outputs the day of occurrence and the value of the maximum temperature via a PLOT block.



1.96 Block MAXXXYP

The MAXXXYP block finds the maximum of its second input signal over a number of steps as defined through its parameter.



Name	MAXXXYP
Function	fb0074
Inputs	2
Outputs	2
Parameters	1
Strings	0
Group	I

Inputs

1 Any x-coordinate
2 Corresponding y-coordinate

Outputs

1 x-coordinate of maximum
2 Corresponding maximum value

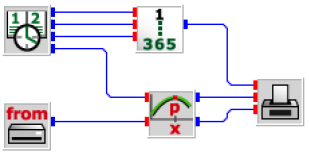
Parameters

1 Number p of steps to find the maximum

Strings

None

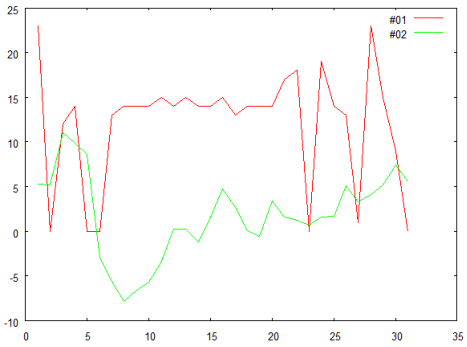
maxxxyp.vseit



A CLOCK block runs through the hours of January 1982. Corresponding temperature data are read by a READ block from file `meteo82.dat` which can be found in the `examples/data` directory.

The MAXXXYP block detects the daily maximum values and outputs the hour of occurrence and the value of the maximum temperature via a PLOT block.

As could be expected mostly the maxima are in the afternoon hours. However, sometimes the maximum is during the night. This happens when a daily temperature shift occurs.



1.97 Block MBE

The MBE block calculates the mean bias error between the input signals over a complete simulation run.

Name	MBE
Function	fb0075
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	I

Inputs

1 Simulated time series x_s
2 Measured time series x_m

Outputs

1 Mean bias error

Parameters

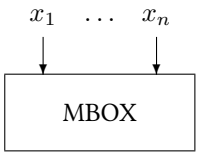
None

Strings

None

1.98 Block MBOX

The MBOX block displays its inputs in a message box.



Format, Title

Name	MBOX
Function	fb0063
Inputs	0 . . . [6]
Outputs	0
Parameters	0
Strings	0 . . . [2]
Group	S

Inputs

1	Any xalue x_1
2	Any xalue x_2
n	Any xalue x_n with $n \leq 6$

Outputs

None

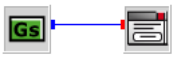
Parameters

None

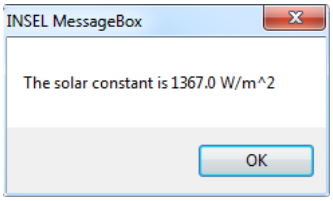
Strings

1	Fortran format for the output, default is the star format '*'
2	Title of the message box

mbox.vseit

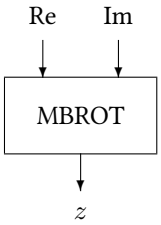


This MBOX block displays the solar constant in a message box.



1.99 Block MBROT

The MBROT block calculates the Mandelbrot set.



Name	MBROT
Function	fb0064
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 x coordinate, i. e., real part
2 y coordinate, i. e., imaginary part

Outputs

1 Mandelbrot color

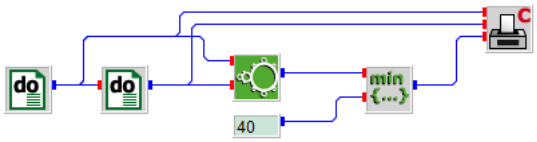
Parameters

None

Strings

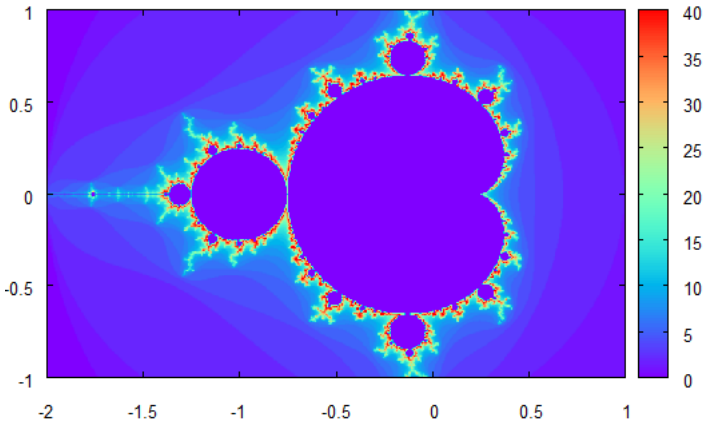
None

mbrot.vseit



An outer DO block varies the real part of a complex variable between -2 and $+1$ in steps of 0.005. The inner DO block varies the imaginary part between -1 and $+1$ in steps of 0.005. Both outputs are connected to an MBROT block whose nice carpet plot is displayed by a PLOTC block.

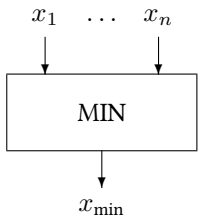
The MIN block restricts the color scale to 40 for representational reasons.



In Germany the Mandelbrot set is known as ”Apfelmännchen”.

1.100 Block MIN

The MIN block returns the minimum of its inputs.



Name	MIN
Function	fb0002
Inputs	1 . . . [999]
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Any value x_1
2	Any value x_2
n	Any value x_n

Outputs

1	Minimum $x_{\min}$ of all x_i for $i = 1 \ldots n \leq 999$

Parameters

None

Strings

None

Description The minimum $x_{\min}$ is calculated from the inputs $x_1 \ldots x_n$ according to

$$x_{\min} = \min\{x_i | i = 1 \ldots n\}$$

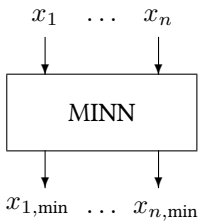
min.vseit

Two CONST blocks provide the constants 1 and 3.14. The MIN blocks checks, which of the inputs is smaller. The SCREEN block displays the result.

```
1.00
```

1.101 Block MINN

The MINN block finds its minimum input signal over a complete simulation run.



Name	MINN
Function	fb0021
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	0
Strings	0
Group	I

Inputs

i Any signal x_i

Outputs

i Minimum over all x_i

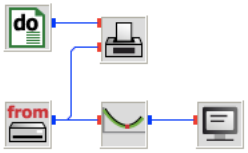
Parameters

None

Strings

None

minn.vseit

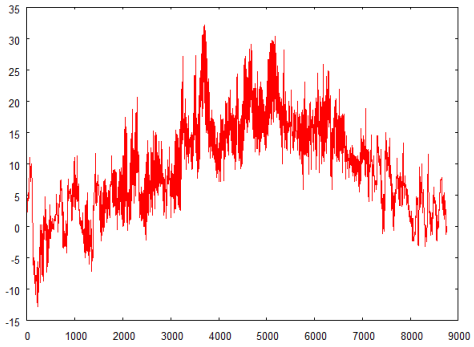


A DO block runs through one year in steps of one hour and controls reading the data file `meteo82.dat` from the `examples/data` directory as specified by the parameters of a READ block. By definition of the Format parameter of the READ block `(33X,F5.1,26X)` only colums 34–38 are evaluated. These contain ambient temperature data in degrees Celsius.

The MINN block finds the global minimum temperature which is finally displayed by a SCREEN block.

```
-12.9
```

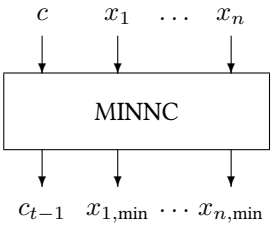
A PLOT block is used to display the complete time series.



See also Block MAXX.

1.102 Block MINNC

The MINNC block scans its second input signal for a minimum as long as the condition input remains constant. When this input changes its value, the minimum is on output and the successors of the block are executed.



Name	MINNC
Function	fb0037
Inputs	2 . . . [51]
Outputs	2 . . . [51]
Parameters	0
Strings	0
Group	I

Inputs

1	Condition input c
2	Any signal x_1
n	Any signal x_{n-1}

Outputs

1	Condition input c delayed by one step
2	Minimum $\min_1$ over all $c =$ const calls
n	Minimum $\min_{n-1}$ over all $c =$ const calls

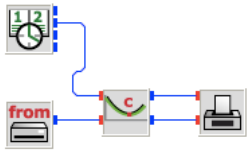
Parameters

None

Strings

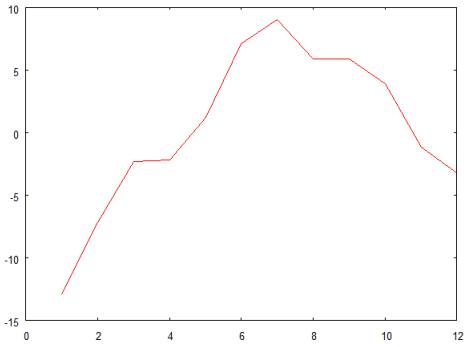
None

minnc.vseit



A CLOCK block runs through one year in steps of one hour and controls reading the data file `meteo82.dat` from the `examples/data` directory as specified by the parameters of a READ block. By definition of the Format parameter of the READ block `(33X,F5.1,26X)` only colums 34–38 are evaluated. These contain ambient temperature data in degrees Celsius.

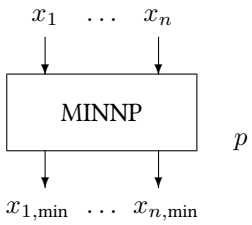
Since the month output of the CLOCK block is used as condition input to the MINNC block, monthly minima of the temperature data are calculated by the MINNC block and finally displayed as graph by the PLOT block.



See also Block MAXXC.

1.103 Block MINNP

The MINNP block finds the minimum of its input signal over a number of steps as defined through its parameter.



Name	MINNP
Function	fb0020
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	1
Strings	0
Group	I

Inputs

i Any signal x_i

Outputs

i Minimum $x_{i,\min}$ of x_i

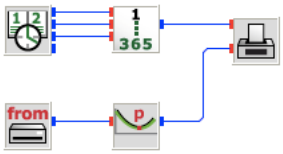
Parameters

1 Number p of steps to find the minimum $x_{i,\min}$

Strings

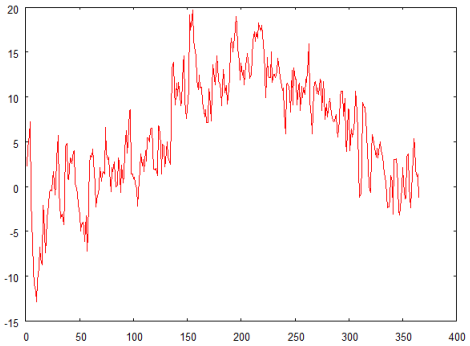
None

minnp.vseit



A CLOCK block runs through one year in steps of one hour and controls reading the data file `meteo82.dat` from the `examples/data` directory as specified by the parameters of a READ block. By definition of the Format parameter of the READ block `(33X,F5.1,26X)` only colums 34–38 are evaluated. These contain ambient temperature data in degrees Celsius.

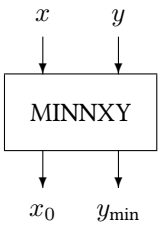
Since the parameter of the MINNP block is set to 24, daily minima of the temperature data are calculated by the MINNP block and finally displayed as graph by the PLOT block.



See also Block MAXXP.

1.104 Block MINNXY

The MINNXY block finds the minimum of its second input over a complete simulation run and returns the x and y coordinates.



Name	MINNXY
Function	fb0070
Inputs	2
Outputs	2
Parameters	0
Strings	0
Group	I

Inputs

1 Any signal x
2 Corresponding value $y(x)$

Outputs

1 x_0 coordinate of minimum
2 Minimum value $y_{min} = y(x_0)$

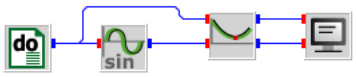
Parameters

None

Strings

None

minnxy.vseit

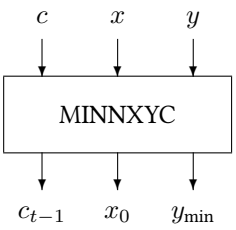


A DO block provides valaues for an angle α between 0 and 360 degrees in steps of one, which are used by a SIN block to calculate $\sin(\alpha)$.

The MINNXY block detects the minimum value and outpus the coordinate of the minimum and the value of the minimum to a SCREEN block to show the result.

```
270.00000      -1.0000000
```

1.105 Block MINNXYC



Name	MINNXYC
Function	fb0073
Inputs	3
Outputs	3
Parameters	0
Strings	0
Group	I

Inputs

1 Condition input c
2 Any signal x
3 Corresponding signal y

Outputs

1 Condition input c delayed by one step
2 x-oordinate of minimum value
3 Minimum over all $c =$ const calls

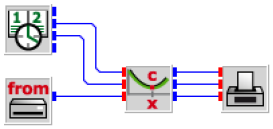
Parameters

None

Strings

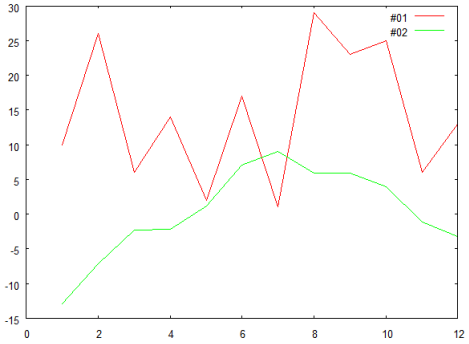
None

minnxyc.vseit



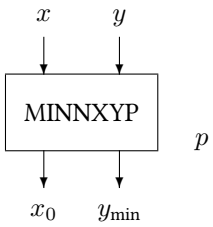
A CLOCK block runs through the hours of the year 1982. Corresponding temperature data are read by a READ block from file `meteo82.dat` which can be found in the `examples/data` directory.

The MINNXYC block detects the monthly minimum values and outputs the day of occurrence and the value of the minimum temperature via a PLOT block.



1.106 Block MINNXYP

The MINNXYP block finds the minimum of its second input signal over a number of steps as defined through its parameter.



Name	MINNXYP
Function	fb0074
Inputs	2
Outputs	2
Parameters	1
Strings	0
Group	I

Inputs

1 Any x-coordinate
2 Corresponding y-coordinate

Outputs

1 x-coordinate of minimum
2 Corresponding minimum value

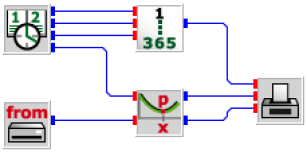
Parameters

1 Number p of steps to find the minimum

Strings

None

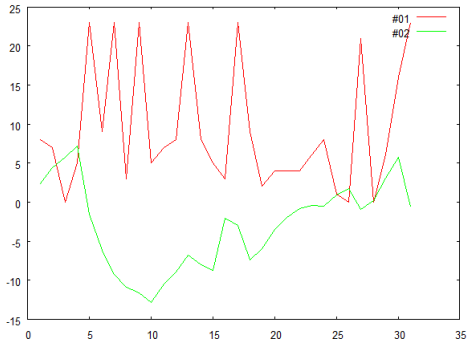
minnxyp.vseit



A CLOCK block runs through the hours of January 1982. Corresponding temperature data are read by a READ block from file `meteo82.dat` which can be found in the `examples/data` directory.

The MINNXYP block detects the daily minimum values and outputs the hour of occurrence and the value of the minimum temperature via a PLOT block.

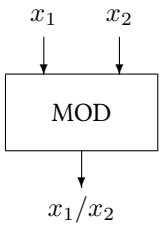
As can be observed by far not all minima are in the early morning hours as might have been expected.



1.107 Block MOD

The MOD block calculates first input modulo the second.

x%y



Name	MOD
Function	fb0004
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Dividend x_1
2 Divisor x_2

Outputs

1 x_1 modulo x_2

Parameters

None

Strings

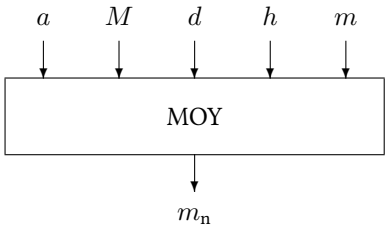
None

Remarks

If the second input value x_2 is zero, the block displays a warning message and returns NaN. If this happens more than once, the warning message is suppressed. At the end of a simulation run the total number of occurencies is displayed.

1.108 Block MOY

The MOY block returns the minute of the year.



Name	MOY
Function	fb0023
Inputs	5
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Year a
2	Month M
3	Day d
4	Hour h
5	Minute m

Outputs

1	Minute of the year $m_n \in [1, 527\,040]$

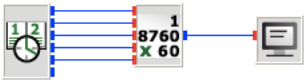
Parameters

None

Strings

None

moy.vseit

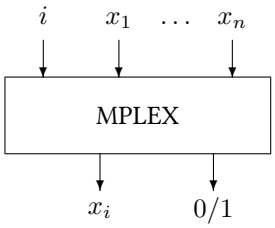


A CLOCK block provides the last few minutes of the year 2010. Date and time are converted into the minute of the year by the MOY block. The result is displayed by the SCREEN block

```
525597.
525598.
525599.
525600.
```

1.109 Block MPLEX

The MPLEX block simulates a multiplexer, i.e. it connects a certain input to the output signal.



Name	MPLEX
Function	fb0048
Inputs	2 . . . [101]
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1	Output index i
2	Any signal x_1
$i+1$	Any signal x_i
$n+1$	Any signal x_n

Outputs

1	Multiplexed signal x_i, when $1 \leq \mathrm{i} \leq n$, else set to 0
2	Error indicator is set to 0 if $1 \leq \mathrm{i} \leq n$, else set to 1.

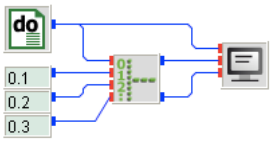
Parameters

None

Strings

None

mplex.vseit

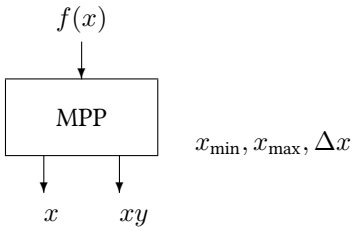


A DO counts from zero to four, used as first input of the MPLEX block. Since three CONST blocks are connected to the MPLEX block zero and four are unallowed values for the output index. Therefore, warning messages occur and the SCREEN block displays

```
W05107 Block 00002: Invalid index input
  0.0  0.0  1.0
  1.0  0.1  0.0
  2.0  0.2  0.0
  3.0  0.3  0.0
  4.0  0.0  1.0
W05108 Block 00002: Number of invalid indexes:        2
```

1.110 Block MPP

The MPP block simulates an ideal maximum power point tracker. In general, the MPP block can be used to find the maximum of any unimodal function.



Name	MPP
Function	fb0057
Inputs	1
Outputs	2
Parameters	3
Strings	0
Group	L

Inputs

1 Any signal $y = f(x)$ within the loop

Outputs

1 Signal x which is returned to a preceeding block until the product xy reaches its maximum
2 Product xy

Parameters

1 Lower limit $x_{\min}$ of the interval in which the iteration is performed
2 Upper limit $x_{\max}$ of the iteration intervall
3 Error tolerance Δx for the accuracy of the calculated maximum

Strings

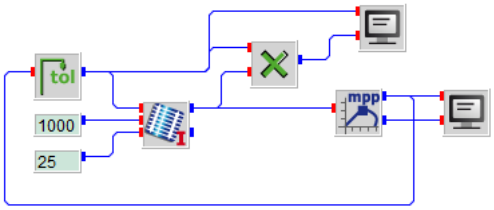
None

Description

The standard application of this block is the calculation of maximum-power- points of a photovoltaic generator under different radiation and temperature conditions. In this case the MPP block determines the maximum product of the naturally unimodal function $P = V \cdot I$ for a given $I(V)$ characteristic in the intervall $[V_{\min}, V_{\max}]$ as given by the parameters.

Please observe that the input $f(x)$ of the MPP block comes back to MPP as a result of the output of the MPP block x and that MPP maximizes the product $x \cdot f(x)$ and not the function $f(x)$ itself.

mpp.vseit



A PVI block is used to calculate the current of a PV module at Standard Test Conditions, i. e., global irradiance equal to 1000 W/m^2 and a module temperature of 25 degrees Celsius, as specified by two CONST blocks.

The output current of the PV module is connected to an MPP block, which iterates the module voltage between 0 and 30 V, as specified by the parameters. The error tolerance of the MPP block is set to 0.01 V, which means that the block stops the iteration when the voltage interval decreases to a value less than 0.01 V.

The output voltage of the MPP block is connected to a TOL block whose task it is to provide the input voltage to the PVI block in this case. The iteration can be observed by the SCREEN block which is connected to the TOL block.

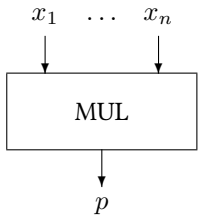
When the iteration is finished, i. e., the voltage iteration interval shrank below 0.01 V the MPP block gives control to its successors, which is a SCREEN block in this case. Finally, the SCREEN block displays the DC power at the maximum-power point of the module

```
11.459    37.640
18.541    49.531
22.918     0.000
15.836    50.642
14.164    46.121
16.869    52.374
17.508    52.467
17.902    51.901
17.264    52.556
17.113    52.530
17.357    52.543
17.206    52.553
```

```
    17.299    52.554
    17.242    52.556
    17.277    52.556
    17.255    52.556
    17.250    52.556
    17.259    52.556
Maximum at 17.259 Volt and 52.556 Watt
```

1.111 Block MUL

The MUL block multiplies its inputs.



Name	MUL
Function	fb0002
Inputs	1 . . . [999]
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Any value x_1
2	Any value x_2
n	Any value x_n

Outputs

1	Product p of all x_i for $i = 1 \ldots n \leq 999$

Parameters

None

Strings

None

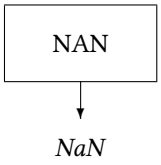
mul.vseit

INSEL knows the multiplication tables which start with one times one is equal to one. A CONST blocks defines the value one. The MUL block multiplies one by one and the SCREEN block displays the result.

```
1.0
```

1.112 Block NAN

The NAN block returns *Not A Number*, which could be useful to indicate missing data in plots.



Name	NAN
Function	fb0003
Inputs	0
Outputs	1
Parameters	0
Strings	0
Group	C

Inputs

None

Outputs

1 *Not A Number*. Any computation based on this value will also be *Not a Number*. Can be used in conjunction with MPLEX.

Parameters

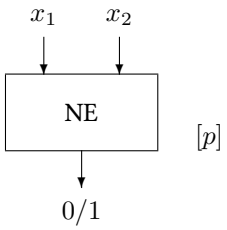
None

Strings

None

1.113 Block NE

The NE block tests whether its first input is different from the second one.



Name	NE
Function	fb0027
Inputs	2
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Any value x_1
2 Any value x_2

Outputs

1 y is set to one if $|x_1 - x_2| > p$, otherwise y is set to zero

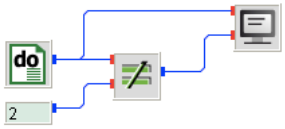
Parameters

1 Error tolerance $p \geq 0$ for the difference between x_1 and x_2, set to zero by default

Strings

None

ne.vseit

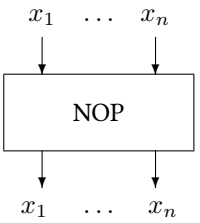


A DO block counts from 1 to 3. The CONST block provides a constant with value 2. The NE blocks checks, whether the output of the DO block is not equal to 2 (true = 1) or equal (false = 0). The SCREEN block displays the result.

```
1.0   1.0
2.0   0.0
3.0   1.0
```

1.114 Block NOP

The NOP block performs no operation except setting the output value equal to the input.



Name	NOP
Function	fb0005
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	0
Strings	0
Group	S

Inputs

i Any value x_i

Outputs

i Value of x_i

Parameters

None

Strings

None

nop.vseit

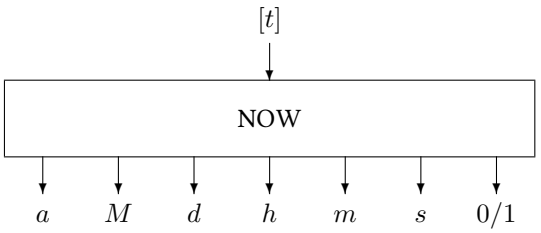
The CONST block provides a constant with value 1. The NOP blocks passes its input through and the SCREEN block displays the result.

```
1.0
```

1.115 Block NOW

The NOW block returns the current date and time from the operating system.

(now)



Name	NOW
Function	fb0050
Inputs	0 . . . [1]
Outputs	7
Parameters	0
Strings	0
Group	S

Inputs

1	Any predecessor

Outputs

1	Current year
2	Current month
3	Current day
4	Current hour
5	Current minute
6	Current second
7	Daylight saving time

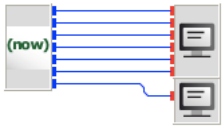
Parameters

None

Strings

None

now.vseit



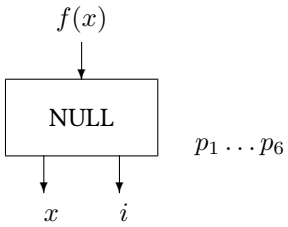
The NOW block provides the current date, time and daylight-saving-time indicator which are displayed by two SCREEN blocks.

```
2013.000      9.000     21.000     13.000     14.000     36.201
   1.
```

Of course, the output depends on the moment when the program is executed.

1.116 Block NULL

The NULL block searches a root of a continuous function.



Name	NULL
Function	fb0054
Inputs	1
Outputs	2
Parameters	6
Strings	0
Group	L

Inputs

1 Signal $y = f(x)$, which corresponds to the output x

Outputs

1 Signal x which is varied iteratively until $y = 0 \pm \Delta y_{\max}$. This output has to be connected to a corresponding TOL block.

2 Indicator i for iteration failure
- 0 Solution found
- 1 Too many iterations
- 2 Both function values positive at boundaries
- 3 Both function values negative at boundaries
- 4 Found trivial solution

Parameters

1 Mode
- 0 Involution algorithm (in the current version this is the only option)

2 Lower limit $x_{\min}$ of the iteration interval

3 Upper limit $x_{\max}$ of the iteration intervall

4 Tolerance $\Delta y_{\max}$ for the accuracy of the calculated root

5 Maximum number $N_{\max}$ of iterations

6 Output value for x which is returned when the number N_{max} of iterations is reached or no solution was found within the iteration interval

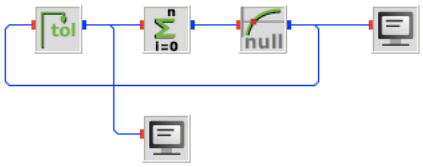
Strings

None

Description

The block determines the root of a function by different algorithms which can be set via the block's first parameter. The output x is changed iteratively until y is equal to $0 \pm \Delta y_{max}$.

null.vseit



A POLYN block provides the function $y = x^2 - 1$. The NULL block in combination with the TOL block finds the root of the function by the involution algorith in the interval $[0, 5]$. The SCREEN block, which is connected directly with the output off the TOL block displays all iteration steps. The SCREEN block which is connected to the NULL block displays the solution only.

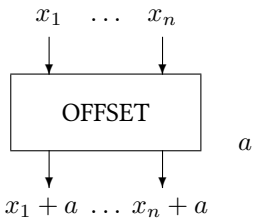
```
0.0000000000
5.0000000000
2.5000000000
1.2500000000
0.6250000000
0.9375000000
1.0937500000
1.0156250000
0.9765625000
0.9960937500
Solution x = 0.9960937500
```

Remarks

The iteration intervall may contain only a single root.

1.117 Block OFFSET

The OFFSET block adds its input to its parameter.



Name	OFFSET
Function	fb0006
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	1
Strings	0
Group	S

Inputs

i	Any value x_i

Outputs

i	Sum of input and parameter $x_i + a$

Parameters

1	Offset a

Strings

None

offset.vseit

The OFFSET block with parameter 3.14 uses the input of a CONST block with value one. A SCREEN block displays the obvious result.

4.14

1.118 Block OR

The OR block checks whether its first input or the second one or both are equal to one (true).

Name	OR
Function	fb0030
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any value x_1 close to one or zero
2 Any value x_2 close to one or zero

Outputs

1 y is set to one if x_1 or x_2 or both are $\in [0.5, 1.5)$, otherwise y is set to zero

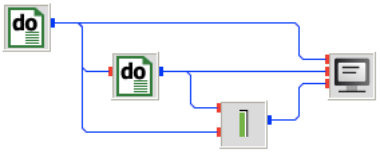
Parameters

None

Strings

None

or.vseit



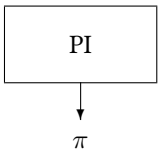
Two nested DO blocks with initial value 0, final value 1 and increment 1 are linked to an OR block. The result is displayed by the SCREEN block.

```
0.0   0.0   0.0
0.0   1.0   1.0
1.0   0.0   1.0
1.0   1.0   1.0
```

See also Blocks AND, XOR.

1.119 Block PI

The PI block provides the pi constant.



Name	PI
Function	fb0003
Inputs	0
Outputs	1
Parameters	0
Strings	0
Group	C

Inputs

None

Outputs

1 Constant π

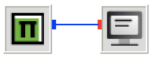
Parameters

None

Strings

None

pi.vseit



The PI block provides the constant π which is displayed by the SCREEN block.

```
3.1415927
```

1.120 Block PID

The PID block simulates a PID controller.

Name	PID
Function	fb0061
Inputs	4
Outputs	6
Parameters	6
Strings	0
Group	S

Inputs

1 Setpoint
2 Control
3 Flag
4 Time

Outputs

1 Control
2 Error
3 Control without anti-windup
4 Prop
5 Int
6 Der

Parameters

1 K
2 Ti
3 Td
4 Min val
5 Max val

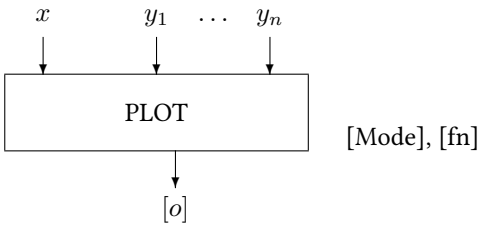
6	Thres	

Strings

None

1.121 Block PLOT

The PLOT block generates graphical output of its connected input data via gnuplot.



Name	PLOT
Function	fb0044
Inputs	2 . . . [20]
Outputs	0 . . . [1]
Parameters	0
Strings	0 . . . [1]
Group	S

Inputs

1	Any signal x
2	Any signal y_1
n	Any signal y_{n-1}

Outputs

1	User defined block number (if available)

Parameters

None

Strings

1	File name fn of a gnuplot command file. If no file name is provided, a default file `insel.gnu` is generated with default gnuplot commands.

Description

The PLOT block uses the open-source software gnuplot to generate graphical output of x-y data. As long as a model is executed the block collects all connected data and writes them to a temporary file named `insel.gpl` (gnuplot data).

Many parameters can be set in the graphical interface, e.g. title, labels, ranges, units, axis (y1 or y2), colors. By default, the plot will be displayed in a window, but it can also be exported to PNG, PDF, SVG, ...

Once the plot is displayed, it is possible to zoom and pan with arrows, mouse or + - keys. The window can be closed with the mouse or by pressing q.

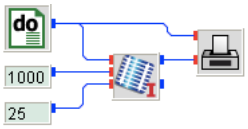
According to the Gnuplot documentation, here are the available commands:

```
<B1> doubleclick   send mouse coordinates to clipboard (pm win wxt x11)
<B2>               annotate the graph using ‘mouseformat‘ (see keys ’1’, ’2’)
                   or draw labels if ‘set mouse labels is on‘
<Ctrl-B2>          remove label close to pointer if ‘set mouse labels‘ is on
<B3>               mark zoom region (only for 2d-plots and maps)
<B1-Motion>        change view (rotation); use <Ctrl> to rotate the axes only
<B2-Motion>        change view (scaling); use <Ctrl> to scale the axes only
<Shift-B2-Motion>  vertical motion -- change xyplane
<B3-Motion>        change view (azimuth)
<wheel-up>           scroll up (in +Y direction)
<wheel-down>         scroll down
<shift-wheel-up>     scroll left (in -X direction)
<shift-wheel-down>   scroll right
<Control-WheelUp>    zoom in on mouse position
<Control-WheelDown>  zoom out on mouse position
<Shift-Control-WheelUp>    pinch on x
<Shift-Control-WheelDown>  expand on x

q            * close this plot window

a               ‘builtin-autoscale‘ (set autoscale keepfix; replot)
b               ‘builtin-toggle-border‘
e               ‘builtin-replot‘
g               ‘builtin-toggle-grid‘
h               ‘builtin-help‘
i               ‘builtin-invert-plot-visibilities‘
l               ‘builtin-toggle-log‘ y logscale for plots, z and cb for splots
L               ‘builtin-nearest-log‘ toggle logscale of axis nearest cursor
m               ‘builtin-toggle-mouse‘
r               ‘builtin-toggle-ruler‘
V               ‘builtin-set-plots-invisible‘
v               ‘builtin-set-plots-visible‘
1               ‘builtin-previous-mouse-format‘
2               ‘builtin-next-mouse-format‘
5               ‘builtin-toggle-polardistance‘
6               ‘builtin-toggle-verbose‘
7               ‘builtin-toggle-ratio‘
n               ‘builtin-zoom-next‘ go to next zoom in the zoom stack
p               ‘builtin-zoom-previous‘ go to previous zoom in the zoom stack
u               ‘builtin-unzoom‘
```

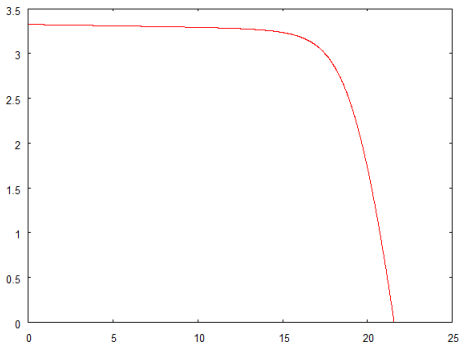
```
+               'builtin-zoom-in' zoom in
=               'builtin-zoom-in' zoom in
-               'builtin-zoom-out' zoom out
Right           'scroll right in 2d, rotate right in 3d'; <Shift> faster
Up              'scroll up in 2d, rotate up in 3d'; <Shift> faster
Left            'scroll left in 2d, rotate left in 3d'; <Shift> faster
Down            'scroll down in 2d, rotate down in 3d'; <Shift> faster
<               'rotate azimuth left in 3d'; <ctrl> faster
>               'rotate azimuth right in 3d'; <ctrl> faster
Escape          'builtin-cancel-zoom' cancel zoom region
```

Limitations Only one PLOT or PLOTP block can be used per INSEL model.

plot.vseit



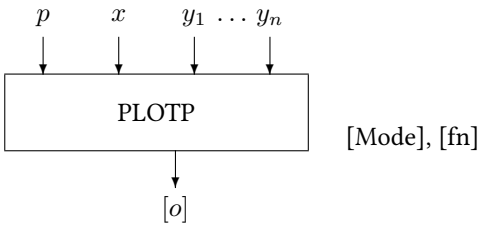
A DO block varies the voltage of a PV module between 0 and 25 Volt in steps of 0.01 Volt. Two CONST blocks provide solar radiation of 1000 Watt per square meter and a module temperature of 25 degrees Celsius, i. e., Standard Test Conditions. A PVI block caluculates the module current. Finally, the PLOT block dispays the I-V characteristic on screen.



See also Block PLOTP.

1.122 Block PLOTP

The PLOTP block generates a parametric graphical output of its connected input data via gnuplot.



Name	PLOTP
Function	fb0044
Inputs	3 . . . [20]
Outputs	0 . . . [1]
Parameters	0
Strings	0 . . . [1]
Group	S

Inputs

1	Curve parameter p
2	Any signal x
3	Any signal y_1
n	Any signal y_{n-2}

Outputs

1	User defined block number (if available)

Parameters

None

Strings

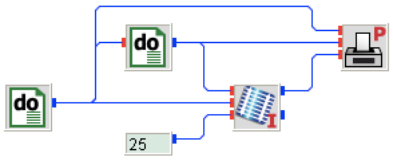
1	File name fn of a gnuplot command file. If no file name is provided, a default file `insel.gnu` is generated with default gnuplot commands.

Description The PLOTP block is nearly identical with the PLOT block. The only difference is that the PLOTP block has an additional input. The first input (the parameter input) is used to distinguish between several logical curves. As long as this input is constant, all data points are assumed to belong to the same curve. When the first input changes its value, a pen-up pen-down movement is simulated and a new curve drawn.

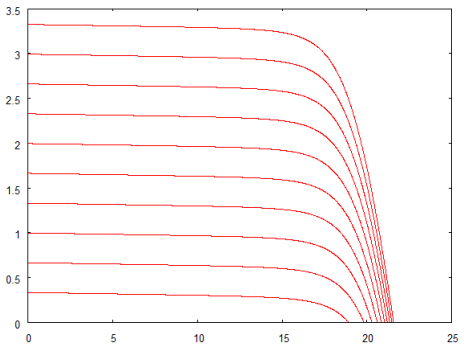
The second input becomes the x coordinate. All other inputs can take up y data.

Limitations Only one PLOT block or one PLOTP block can be used in an INSEL model file.

plotp.vseit This is an example how to plot a set of I-V curves of a photovoltaic module for different irradiance levels.



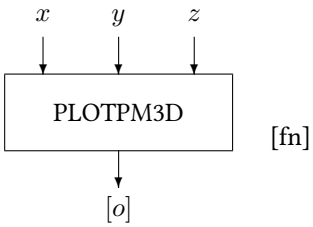
Two nested DO blocks vary the radiation of a PV module between 100 and 1000 Watt per square meter in steps of 100 Watt per square meter (outer DO loop) and voltage between 0 and 25 Volt in steps of 0.01 Volt (inner DO loop). A CONST block provides a module temperature of 25 degrees Celsius. A PVI block calculates the module current. Finally, the PLOTP block displays the I-V characteristics on screen.



See also Block PLOT.

1.123 Block PLOTPM3D

The PLOTPM3D block generates a palette-mapped 3D plot output of its connected input data via gnuplot.



Name	PLOTPM3D
Function	fb0044
Inputs	3
Outputs	0 . . . [1]
Parameters	0
Strings	0 . . . [1]
Group	S

Inputs

1 Any signal x
2 Any signal y
3 Any signal z

Outputs

1 User defined block number (if available)

Parameters

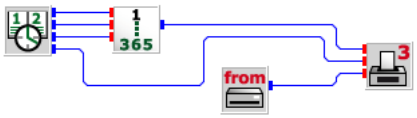
None

Strings

1 File name fn of a gnuplot command file. If no file name is provided, a default file `insel.gnu` is generated with default gnuplot commands.

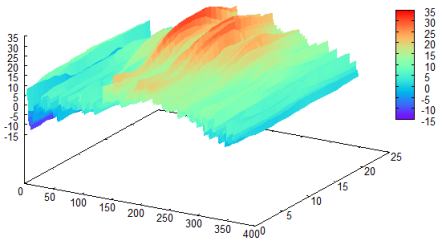
Limitations Only one PLOTPM3D block or one PLOTPPMC block can be used in an INSEL model file.

plotpm3d.vseit



A READ block reads one complete year of temperature from the data file `temperature.dat` located in the `examples/data` in hourly resolution. The time steps are provided by a CLOCK block. The DOY block converts the date and time outputs of the CLOCK block into the day of the year which is used as x input of the PLOTPM3D block. The y coordinate is the hour output of the CLOCK block. The z coordinate is connected to the READ block.

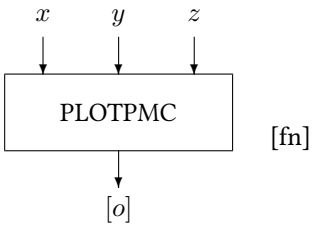
Finally, the PLOTPM3D block displays the temperature data on screen.



See also Block PLOTPMC.

1.124 Block PLOTPMC

The PLOTPMC block generates a heatmap (a.k.a. carpet plot) of its connected input data via gnuplot.



Name	PLOTPMC
Function	fb0044
Inputs	3
Outputs	0 . . . [1]
Parameters	0
Strings	0 . . . [1]
Group	S

Inputs

1 Any signal x
2 Any signal y
3 Any signal z

Outputs

1 User defined block number (if available)

Parameters

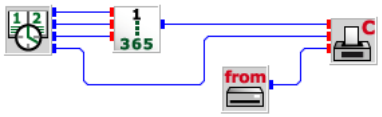
None

Strings

1 File name fn of a gnuplot command file. If no file name is provided, a default file `insel.gnu` is generated with default gnuplot commands.

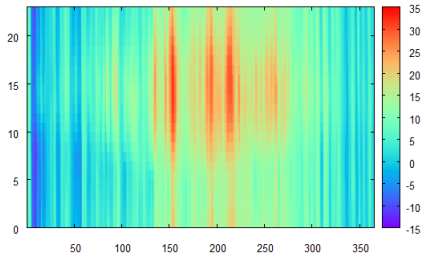
Limitations Only one PLOTPM3D block or one PLOTPMC block can be used in an INSEL model file.

plotpmc.vseit



A READ block reads one complete year of temperature data from file `temperature.dat` located in the `examples/data` in hourly resolution. The time steps are provided by a CLOCK block. The DOY block converts the date and time outputs of the CLOCK block into the day of the year which is used as x input of the PLOTPMC block. The y coordinate is the hour output of the CLOCK block. The z coordinate is connected to the READ block.

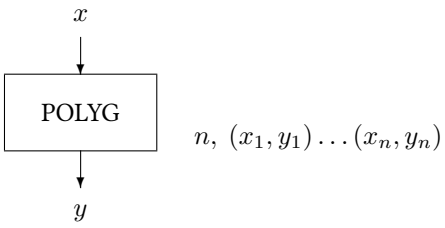
Finally, the PLOTPMC block displays the temperature data on screen.



See also Block PLOTPM3D.

1.125 Block POLYG

The POLYG block interpolates a polygon of points defined through the parameters.



Name	POLYG
Function	fb0019
Inputs	1
Outputs	1
Parameters	3 . . . [7001]
Strings	0
Group	S

Inputs

1 Any signal x

Outputs

1 Interpolated value $y(x)$

Parameters

1 Number of given points, $2 \leq n \leq 3500$
2 x-coordinate of 1st point, x_1
3 y-coordinate of 1st point, y_1
4 x-coordinate of 2nd point, x_2
5 y-coordinate of 2nd point, y_2
. . .
$2n$ x-coordinate of n^{th} point, x_n
$2n+1$ y-coordinate of n^{th} point, y_n

Strings

None

Description

The x-coordinates $x_1 \ldots x_n$ must either be stricty increasing, i. e., $x_1 < x_2 \ldots < x_n$ or strictly decreasing, $x_1 > x_2 \ldots > x_n$. The output signal y is calculated from the input x according to

$$y = \begin{cases} y_1 & \text{if } x < x_1 \\ y_i + (x - x_i)(y_{i+1} - y_i)/(x_{i+1} - x_i) & \text{if } x_i \le x < x_{i+1} \\ y_n & \text{if } x_n < x \end{cases}$$

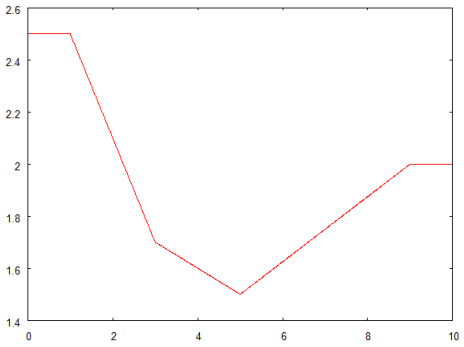
for the case of increasing x values and correspondingy for decreasing x values.

polyg.vseit

A DO block delivers values from 0 to 10 in steps of 0.1. The POLYG block interpolates as specified by the block parameters

```
4
1 2.5
3 1.7
5 1.5
9 2
```

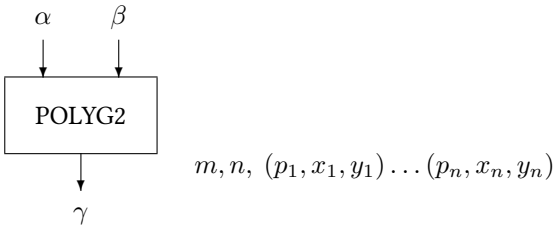
The PLOT block displays the graph.



Please observe how the POLYG block keeps the output constant for data out of the specified parameter range.

1.126 Block POLYG2

The POLYG2 block interpolates on a two dimensional polygon of points as defined by parameters.



Name	POLYG2
Function	fb0049
Inputs	2
Outputs	1
Parameters	8 . . . [1502]
Strings	0
Group	S

Inputs

1 Any signal α
2 Any signal β

Outputs

1 Interpolated value γ

Parameters

1 Interpolation mode m, determines which of the parameters defining the polygon is interpreted as dependant variable γ
- 0 $\gamma = y(p, x) = y(\alpha, \beta)$
- 1 $\gamma = p(x, y) = p(\alpha, \beta)$
- 2 $\gamma = x(p, y) = x(\alpha, \beta)$

2 Number n of given data triples, $2 \leq n \leq 200$
3 Parameter p_1 of 1st point
4 x-coordinate x_1 of 1st point
5 y-coordinate y_1 of 1st point
6 Parameter p_2 of 2nd point
7 x-coordinate x_2 of 2nd point

8	y-coordinate y_2 of 2nd point
...	
$3n$	Parameter p_n of n^{th} point
$3n$+1	x-coordinate x_n of n^{th} point
$3n$+2	y-coordinate y_n of n^{th} point

Strings

None

Remarks

The parameters p_i and x_i must be given in rising order, i.e.

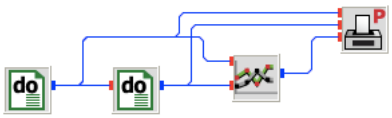
$$\begin{aligned} p_1 &\leq p_2 \leq \ldots \leq p_\text{n} \\ x_1 &\leq x_2 \leq \ldots \leq x_\text{n} \end{aligned}$$

Description

The block performs a two-dimensional linear interpolation on a set of polygons containing n triples $(p_\text{i}, x_\text{i}, y_\text{i})$ where γ is calculated as the output value for given input values (α, β)

In a first step, the block searches four polygon-points neighboured to the input point characterized by (α, β). These points lie on the neighboured polygons which are characterized by the parameters p_k and p_{k+1}. Now the block computes (β, y_A) and (β, y_B) for the points A and B with the linear interpolation used also in block POLYG. Now the required γ value is given by linear interpolation as $\gamma = ((\alpha - p_\text{k})/\Delta p_{\text{k,k+1}})\Delta y_\text{P}$.

polyg2.vseit

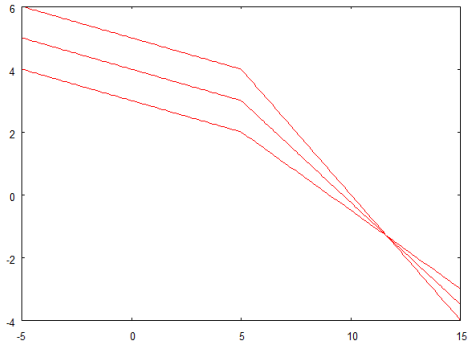


Two nested DO block deliver values between 100 and 200 in steps of 50 (outer DO block) and from -5 to 15 in steps of 0.1 (inner DO block). The POLYG2 block interpolates as specified by the parameters

```
100  0 5
100  5 4
100 10 0
200  0 3
200  5 2
200  9 0
```

The mode is set to $y(p, x) = y(a, b)$ (mode zero) and the number of nodes is equal to six.

The PLOTP block displays the graph.

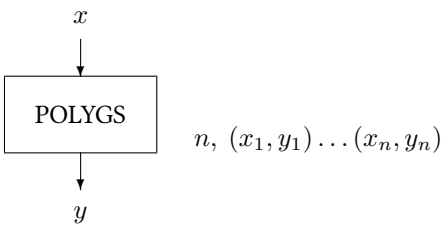


Please observe that – in contrast to the POLYG block – the POLYG2 block extrapolates for data out of the specified parameter range.

See also Block POLYG

1.127 Block POLYGS

The POLYGS block simulates a step function defined through the parameters.



Name	POLYGS
Function	fb0019
Inputs	1
Outputs	1
Parameters	3 . . . [7001]
Strings	0
Group	S

Inputs

1 Any signal x

Outputs

1 Interpolated value $y(x)$

Parameters

1 Number of given points, $2 \leq n \leq 3500$
2 x-coordinate of 1st point, x_1
3 y-coordinate of 1st point, y_1
4 x-coordinate of 2nd point, x_2
5 y-coordinate of 2nd point, y_2
. . .
$2n$ x-coordinate of n^{th} point, x_n
$2n + 1$ y-coordinate of n^{th} point, y_n

Strings

None

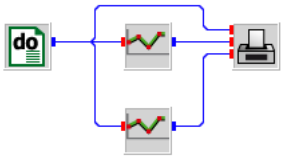
Description

The x-coordinates $x_1 \ldots x_n$ must either be stricty increasing, i. e., $x_1 < x_2 \ldots < x_n$ or strictly decreasing, $x_1 > x_2 \ldots > x_n$. The output signal y is calculated from the input x according to

$$y = \begin{cases} y_1 & \text{if } x < x_1 \\ y_i + (x - x_i)(y_{i+1} - y_i)/(x_{i+1} - x_i) & \text{if } x_i \le x < x_{i+1} \\ y_n & \text{if } x_n < x \end{cases}$$

for the case of increasing x values and correspondingy for decreasing x values.

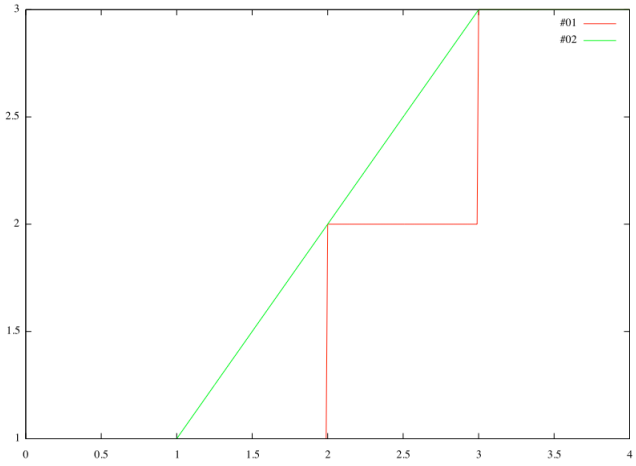
polygs.vseit



A DO block delivers values from 0 to 10 in steps of 0.1. The POLYG block interpolates as specified by the block parameters

```
4
1 2.5
3 1.7
5 1.5
9 2
```

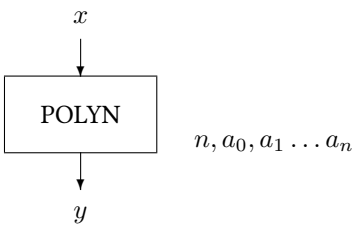
The PLOT block displays the graph.



Please observe how the POLYG block keeps the output constant for data out of the specified parameter range.

1.128 Block POLYN

The POLYN block defines a polynom.



Name	POLYN
Function	fb0012
Inputs	1
Outputs	1
Parameters	2 . . . [12]
Strings	0
Group	S

Inputs

1	Any value x

Outputs

1	$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$

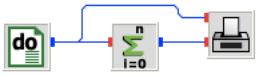
Parameters

1	Order $0 \leq n \leq 10$ of the polynom
2	Coefficient a_0
3	Coefficient a_1
n+2	Coefficient a_n

Strings

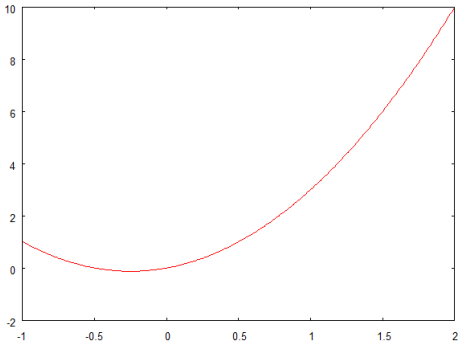
None

polyn.vseit



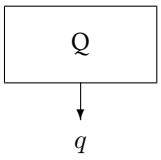
A DO block delivers values from -1 to 2 in steps of 0.01. The POLYN block calculates the polynom $y(x) = 2\,x^2 + x$.

The PLOT block displays the graph.



1.129 Block Q

The Q block provides the elementary charge constant.



Name	Q
Function	fb0003
Inputs	0
Outputs	1
Parameters	0
Strings	0
Group	C

Inputs

None

Outputs

1	Elementary charge q / A s

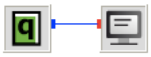
Parameters

None

Strings

None

q.vseit

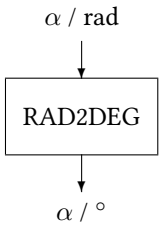


The Q block provides the constant q for the elementary charge of an electron which is displayed by the SCREEN block.

```
1.60218901E-19
```

1.130 Block RAD2DEG

The RAD2DEG block changes radians to degrees.



Name	RAD2DEG
Function	fb0009
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any angle α in radians

Outputs

1 α in degrees

Parameters

None

Strings

None

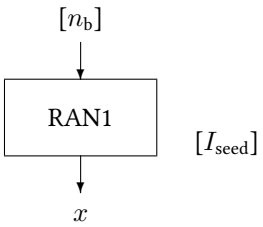
rad2deg.vseit

A PI block provides the constant π which is converted to degrees by the RAD2DEG block to

```
180.00000
```

1.131 Block RAN1

The RAN1 block returns a uniform random deviate between 0.0 and 1.0.



Name	RAN1
Function	fb0032
Inputs	0 . . . [1]
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Block number n_b of a predecessor (optional)

Outputs

1 Uniform random deviate x ∈ [0,1]

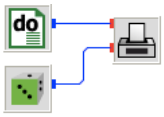
Parameters

1 Initialization I_{seed} of random number generator, set to 1536 by default; the sign of I_{seed} is ignored

Strings

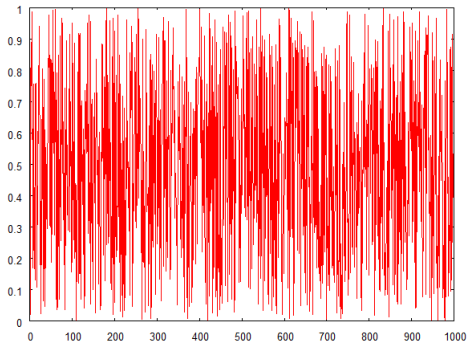
None

ran1.vseit



In a DO 1000 clicks are generated by the parameter setting. For each click the RAN1 block generates one uniformly distributed random number.

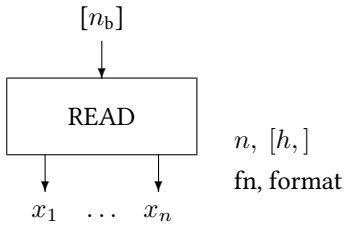
The output of the DO block and the output of the RAN1 block are then plotted by the PLOT block.



See also Block GASDEV.

1.132 Block READ

The READ block reads data from a file with sequential access.



Name	READ
Function	fb0018
Inputs	0 ... [5]
Outputs	1 ... [200]
Parameters	1 ... [2]
Strings	2
Group	S

Inputs

1 Output of a predecessor (optional). If from one time step to the next the value decreases the block performs a rewind of the file.

Outputs

1 Any value x_1 read from file fn
2 Any value x_2 read from file fn
n Any value x_n read from file fn with $n \leq 200$

Parameters

1 Number n of values to be read per record
2 Number h of records to be skipped on the first call (file header, the default is $h = 0$)

Strings

1 File name fn (can contain qualifiers)
2 Fortran format

Remarks

Integers must be read in format Fx.0, where x is the number of bytes.

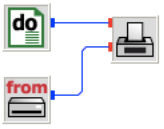
temperature.dat

The next lines show the first ten records of a data file `temperature.dat` which can be found in the `examples/data` directory.

```
4.5
3.8
3.4
3.1
3.0
2.9
3.2
2.8
2.3
2.7
```

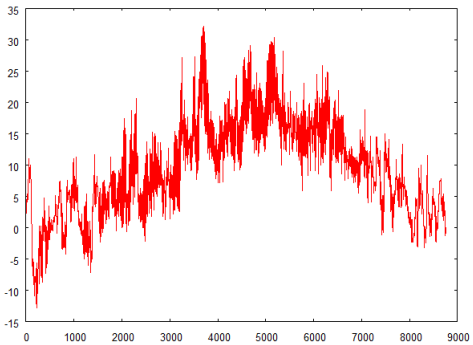
The complete file contains 8760 records of hourly ambient temperature data given in degrees Celsius.

read.vseit



A READ block is used to read the data with sequential access. In this case the general Fortran * format (star format) is used to read the data.

The 'clicks' are generated by a DO block. A PLOT block displays the complete time series.

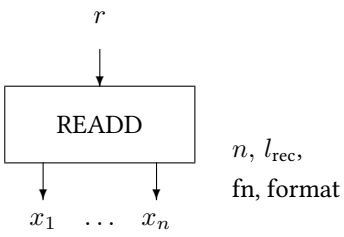


See also

Blocks READD, READN, and WRITE.

1.133 Block READD

The READD reads data from a file with direct access.



Name	READD
Function	fb0016
Inputs	1
Outputs	1 . . . [1000]
Parameters	1 . . . [2]
Strings	2
Group	S

Inputs

1 Record number r

Outputs

1 Any value x_1 read from file fn
2 Any value x_2 read from file fn
n Any value x_n read from file fn with $n \leq 200$

Parameters

1 Number n of values to be read per record
2 Record length l_{rec}

Strings

1 File name fn (can contain qualifiers)
2 Fortran format

Remarks

Integers must be read in format Fx.0, where x is the number of bytes.

All records of file fn must have the same physical record length.

ATTENTION: Different operating systems use different line endings. Windows file records end with two bytes, a carriage return and a line feed character, referred to as CRLF, Unix files have a LF ending, and macOS uses a carriage return character CR. Incompatible line endings result in a read file error message.

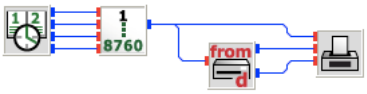
meteo82.dat

The next lines show the first ten records of a data file `meteo82.dat` which can be found in the `examples/data` directory.

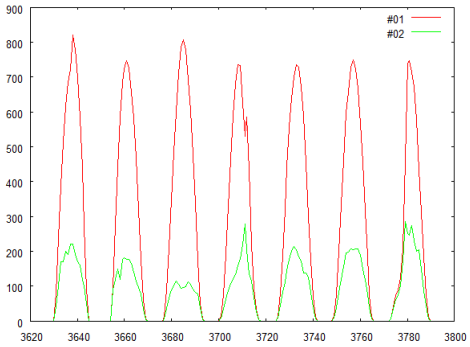
```
 1 182 1   0.   0.   0.   0.   0.  4.5-40.  -40.0-40.0 265.  4.4
 1 182 2   0.   0.   0.   0.   0.  3.8-40.  -40.0-40.0 255.  4.3
 1 182 3   0.   0.   0.   0.   0.  3.4-40.  -40.0-40.0 255.  2.8
 1 182 4   0.   0.   0.   0.   0.  3.1-40.  -40.0-40.0 225.  2.3
 1 182 5   0.   0.   0.   0.   0.  3.0-40.  -40.0-40.0 225.  2.6
 1 182 6   0.   0.   0.   0.   0.  2.9-40.  -40.0-40.0 225.  3.0
 1 182 7   0.   0.   0.   0.   0.  3.2-40.  -40.0-40.0 225.  1.9
 1 182 8   0.   0.   0.   0.   0.  2.8-40.  -40.0-40.0 205.  2.5
 1 182 9   0.   0.   1.   0.   0.  2.3-40.  -40.0-40.0 205.  2.4
 1 18210  15.   6.   6.   3.   0.  2.7-40.  -40.0-40.0 195.  3.0
```

The complete file contains 8760 records of meteorological data, which have been measured in Oldenburg, Germany in the year 1982.

readd.vseit



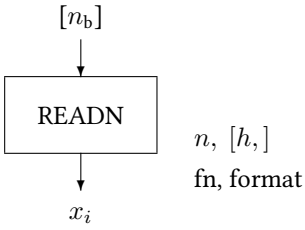
A CLOCK block runs through the first seven days of June, 1982 in steps of one hour as specified by the parameters. The Gregorian date is converted to the hour-of-the-year signal by a HOY block, used as input to plot the hourly data by the PLOT block. The READD block uses the Fortran format `(8X,2F5.0,46X)` in order to read columns 9–14 an 15–20, which contain global and diffuse irradiance data in W/m^2. A PLOT block is used to display the data.



See also Blocks READ, READN, and WRITE.

1.134 Block READN

The READN reads n data from a file with sequential access and successively provides them as outputs; after n calls READN performs a new read.



Name	READN
Function	fb0038
Inputs	0 . . . [5]
Outputs	1
Parameters	2
Strings	2
Group	S

Inputs

1 Output of a predecessor (optional). If from one time step to the next the value decreases the block performs a Rewind of the file.
2 File name qualifier
3 File name qualifier
4 File name qualifier
5 File name qualifier

Outputs

1 Any value x_i read from file fn

Parameters

1 Number n of values to be read per record
2 Number h of records to be skipped on the first call (file header, the default is $h = 0$)

Strings

1 File name fn
2 Fortran format

Remarks

Integers must be read in format Fx.0, where x is the number of bytes.

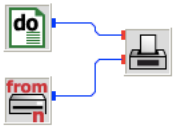
All records of file fn must have the same physical record length.

dwd2002.dat

The next lines show (a part of) the first records of a data file `dwd2002.dat` which can be found in the `examples/data` directory. The file contains irradiance data in kWh per m^2 and month, as published by the German Weather Service DWD for the year 2002.

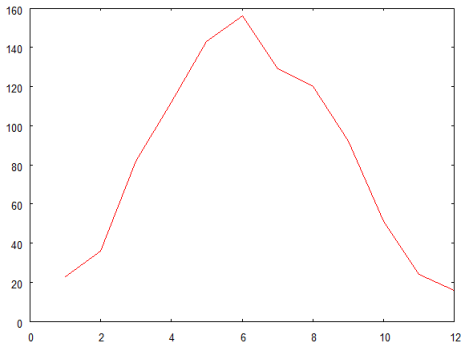
```
Aachen            23   35   88   123   142   ...   16   1001   -1
Augsburg          34   47   100  127   152   ...   22   1178    3
Bad Lippspringe   22   34   79   111   135   ...   16    928   -3
Berlin            21   38   68   103   144   ...   17   1004   -1
Bocholt           20   40   90   119   139   ...   14   1033    3
Bochum            23   36   82   112   143   ...   16    981    2
Bonn              26   41   87   121   142   ...   16   1005    1
Braunlage         20   32   65   104   133   ...   13    938   -2
```

readn.vseit



A READN block is used to read data from the file `dwd2002.dat`. The Number of records to be skipped on the first call is set to five, so that the block actually reads the data for the location Bochum. The number of values to be read per record is set to twelve, with sequential access. In this case the Fortran format `(20X,12F6.0,12X)` is used to read the data.

The 'clicks' are generated by a DO block. A PLOT block displays the twelve monthly means.

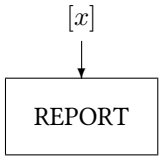


See also

Blocks READ, READD, and WRITE.

1.135 Block REPORT

The REPORT block can be used to generate a primitive report of a simulation run.



Text, unit, type

Name	REPORT
Function	fb0062
Inputs	0 . . . [1]
Outputs	0
Parameters	0 . . . [1]
Strings	0 . . . [2]
Group	S

Inputs

1 Value for a standard line.

Outputs

None

Parameters

1 Type of line
- 0 Standard line (default)
- 1 Headline
- 2 Intermediate headline

Strings

1 Text to be printed
2 Unit of input signal

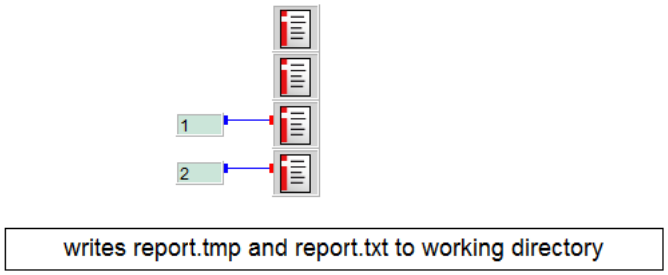
Description

The REPORT block provides a simple way to produce some textual output directly from an INSEL model. Since the blocks can be distributed in several macros, for instance, a sorting mechanism is required so that INSEL can produce ordered output.

This is accomplished by using a four type string followed by an underscore as the first five characters of the text message. This five character code will later be used by INSEL as sorting indicator. After the sorting is completed the five characters will be removed.

The REPORT block writes two data files into the INSEL working directory, a temporary file named `report.tmp` which is unsorted and includes the five leading characters, and the report file `report.txt` which is well formatted.

report.vseit



The example file contains four REPORT blocks. The first two – without any inputs – define two Headlines with the text 0000_Overall Headline and 0010_Intermediate Headline, the second two REPORT blocks define two Standard lines with inputs, which are set to one and two, respectively.

By the way, the text in the `.vseit` file comes from a Headline object as provided by the Objects palette.

report.tmp The unsorted report file becomes

```
(((((INSEL REPORT GENERATOR VERSION 1.3
)))))
)))))
00101
0010_Intermediate Headline
0010a--------------------------------------------------------------------------------
00001
0000_Overall Headline
0000a================
0000b
0030_Second Textline                                    2.00000 Unit 2
0020_First Textline                                     1.00000 Unit 1
```

report.txt After sorting the final report becomes

```
INSEL REPORT GENERATOR VERSION 1.3


Overall Headline
================


Intermediate Headline
--------------------------------------------------------------------------------
First Textline                                          1.00000 Unit 1
Second Textline                                         2.00000 Unit 2
```

1.136 Block RMAE

The RMAE block calculates the relative mean absolute error between the input signals over a complete simulation run.

Name	RMAE
Function	fb0075
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	I

Inputs

1 Simulated time series x_s
2 Measured time series x_m

Outputs

1 Relative mean absolute error

Parameters

None

Strings

None

1.137 Block RMBE

The RMBE block calculates the relative mean bias error between the input signals over a complete simulation run.

Name	RMBE
Function	fb0075
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	I

Inputs

1 Simulated time series x_s
2 Measured time series x_m

Outputs

1 Relative mean bias error

Parameters

None

Strings

None

1.138 Block RMSE

The RMSE block calculates the root mean square error between the input signals over a complete simulation run.

Name	RMSE
Function	fb0075
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	I

Inputs

1 Simulated time series x_s
2 Measured time series x_m

Outputs

1 Root mean square error

Parameters

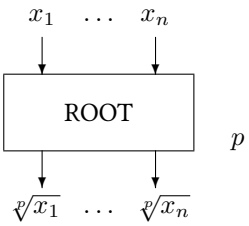
None

Strings

None

1.139 Block ROOT

The ROOT block returns the root of its input as defined through its parameter.



Name	ROOT
Function	fb0006
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	1
Strings	0
Group	S

Inputs

i	Any value $x_i \geq 0$

Outputs

i	p^{th} root $\sqrt[p]{x_i}$

Parameters

1	Root exponent p with $\lvert p \rvert > 1$

Strings

None

Remarks

If the input value x is negative, the ROOT block displays a warning message and does no operation. If this happens more than once, the warning message is suppressed. At the end of a simulation run the total number of occurencies is displayed.

root.vseit

A CONST block provides the value 3.14. A ROOT block with parameter three calculates the third root of its input which is displayed by the SCREEN block

```
1.46434438
```

1.140 Block RRMSE

The RMSE block calculates the relative root mean square error between the input signals over a complete simulation run.

Name	RRMSE
Function	fb0075
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	I

Inputs

1 Simulated time series x_s
2 Measured time series x_m

Outputs

1 Relative root mean square error

Parameters

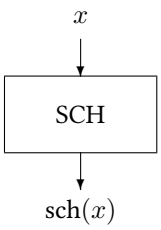
None

Strings

None

1.141 Block SCH

The SCH block returns the hyperbolic secant of its input.



Name	SCH
Function	fb0068
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any signal x

Outputs

1 Hyperbolic secant

Parameters

None

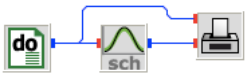
Strings

None

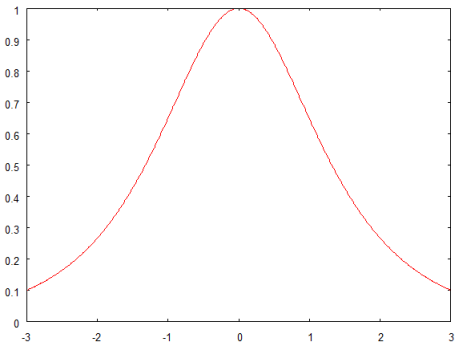
Description The hyperbolic secant is defined as

$$\mathrm{sc}(x) = \frac{2}{\exp(x) + \exp(-x)}$$

sch.vseit

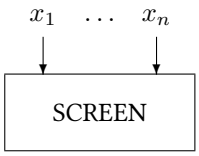


A DO block varies its output from -3 to $+3$ in steps of 0.01. The SCH block calculates hyperbolic secant and the PLOT block displays the graph on screen.



1.142 Block SCREEN

The SCREEN block displays its inputs on the computer display.



[Format, Headline]

Name	SCREEN
Function	fb0014
Inputs	1 . . . [10]
Outputs	0
Parameters	0
Strings	0 . . . [2]
Group	S

Inputs

1	Any xalue x_1
2	Any xalue x_2
n	Any xalue x_n with $n \leq 10$

Outputs

None

Parameters

None

Strings

1	Fortran format for the output, default is the star format '*'
2	Optional headline

screen.vseit

The SCREEN block is used to print

```
hello, world
```

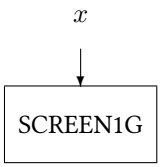
on screen. The parameter of the SCREEN block is set to

```
(''hello, world'')
```

Please observe the use of the quotes: In general, INSEL strings are enclosed by single quotes. When a string shall contain a quote it is common practise to double the quote. Please notice that two single quotes are not equal to a double quote.

1.143 Block SCREEN1G

The SCREEN1G block displays its input directly on a small block, inside the graphical interface. The values are updated automatically during the simulation. When launched from the console, this block doesn't do anything.



Name	SCREEN1G
Function	fb0014
Inputs	1
Outputs	0
Parameters	0
Strings	0
Group	S

Inputs

1 Any xalue x

Outputs

None

Parameters

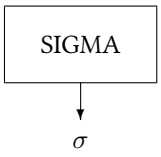
None

Strings

None

1.144 Block SIGMA

The SIGMA block provides the Stefan-Boltzmann constant.



Name	SIGMA
Function	fb0003
Inputs	0
Outputs	1
Parameters	0
Strings	0
Group	C

Inputs

None

Outputs

1 Stefan-Boltzmann constant σ

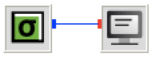
Parameters

None

Strings

None

sigma.vseit

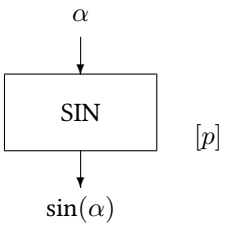


The SIGMA block provides the Stefan-Boltzmann constant which is displayed by the SCREEN block.

```
5.67030014E-08
```

1.145 Block SIN

The SIN block returns the sine of its input.



Name	SIN
Function	fb0007
Inputs	1
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Any angle α

Outputs

1 Sine of α

Parameters

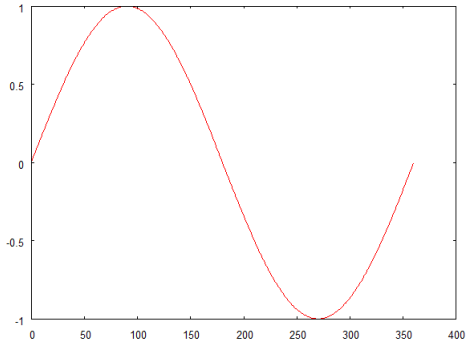
1 Degrees / radians switch p. If p is set to 0 (default) α is interpreted in degrees. If p is set to 1 α is interpreted in radians.

Strings

None

sin.vseit

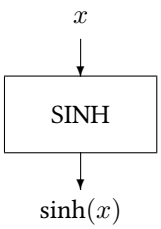
A DO block varies α from 0 to 360 degrees. The SIN block calculates $\sin(\alpha)$ and the PLOT block displays the graph on screen.



See also Blocks ACOS, ACOT, ASIN, ATAN, COS, COT, and TAN.

1.146 Block SINH

The SINH block returns the hyperbolic sine of its input.



Name	SINH
Function	fb0068
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any signal x

Outputs

1 Hyperbolic sine

Parameters

None

Strings

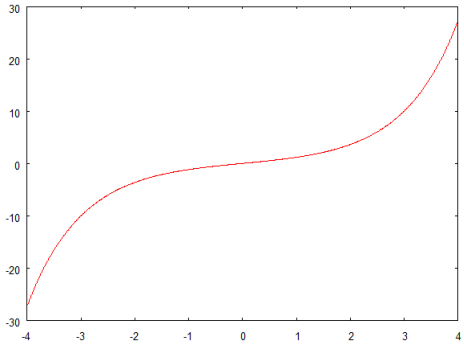
None

Description The hyperbolic sine is defined as

$$\sinh(x) = \frac{\exp(x) - \exp(-x)}{2}$$

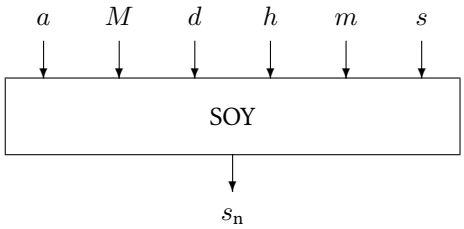
sinh.vseit

A DO block varies x from -4 to $+4$ in steps of 0.01. The SINH block calculates $\sinh(x)$ and the PLOT block displays the graph on screen.



1.147 Block SOY

The SOY block returns the second of the year.



Name	SOY
Function	fb0023
Inputs	6
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Year a
2	Month M
3	Day d
4	Hour h
5	Minute m
6	Second s

Outputs

1	Second of the year $s_n \in [1, 31\,622\,400]$

Parameters

None

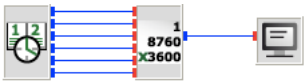
Strings

None

Remarks

Due to the limited accuracy of the Fortran Real*4 format at the end of a year some deviations in the range of 10 seconds may occur.

soy.vseit



A CLOCK block provides the last ten seconds of the year 2010. Date and time are converted into the second of the year by the SOY block. The result is displayed by the SCREEN block

```
31535992.
31535992.
31535992.
31535994.
31535996.
31535996.
31535996.
31535998.
31536000.
31536000.
```

1.148 Block SPEAR

The SPEAR block calculates Spearman's rank correlation coefficient of its inputs over a complete simulation run.

Name	SPEAR
Function	fb0076
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	I

Inputs

1	Any signal x_i
1	Any signal y_i

Outputs

1	Spearman's rank coefficient r_s over all (x_i, y_i)

Parameters

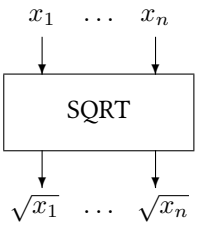
None

Strings

None

1.149 Block SQRT

The SQRT block returns the square root of its input.



Name	SQRT
Function	fb0005
Inputs	1 . . . [10]
Outputs	1 . . . [10]
Parameters	0
Strings	0
Group	S

Inputs

i	Any value $x_i \geq 0$

Outputs

i	Square root $\sqrt{x_i}$

Parameters

None

Strings

None

Remarks

If the input value x is negative, the SQRT block displays a warning message and performs no operation. If this happens more than once, the warning message is suppressed. At the end of a simulation run the total number of occurencies is displayed.

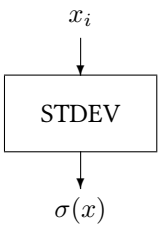
sqrt.vseit

A CONST block defines the number 3.14. The SQRT block returns the square root of its input, which is displayed by the SCREEN block.

```
1.77200449
```

1.150 Block STDEV

The STDEV block calculates the standard deviation of its input signal over a complete simulation run.



Name	STDEV
Function	fb0021
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	0
Strings	0
Group	I

Inputs

i Any signal x_i

Outputs

i Standard deviation σ over all x_i

Parameters

None

Strings

None

Description For the standard deviation σ of a set of numbers $x_i, i = 1 \ldots N$ two slightly different definitions exist in literature.

$$\sigma = \left(\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2 \right)^{1/2}$$

and

$$\sigma = \left(\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2 \right)^{1/2}$$

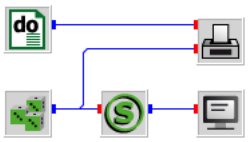
where $\bar{x}$ denotes the average

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

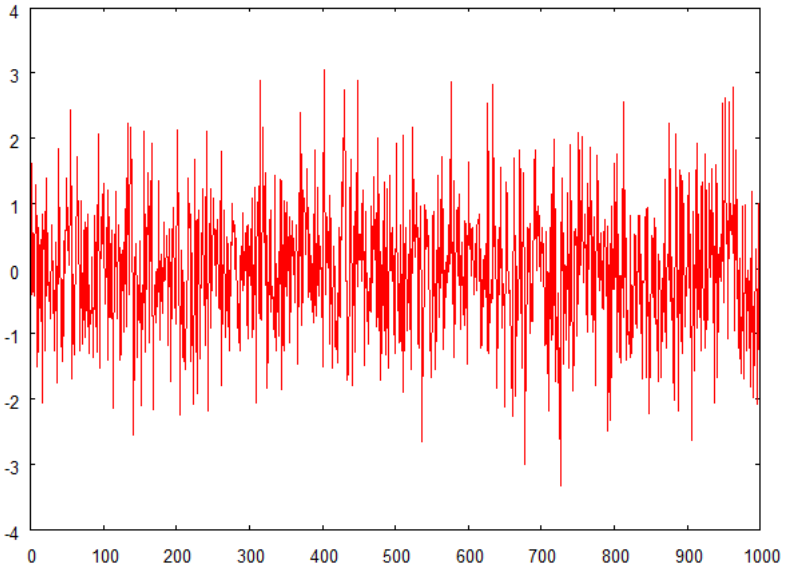
and N the number of elements in the sample.

The first equation is used by the STDEV block.

stdev.vseit



A DO block provides 1000 clicks, so that the GASDEV block generates 1000 normaly distributed random numbers which are displayed by a PLOT block.



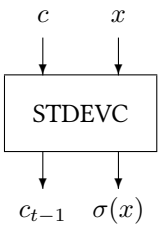
As expected, the standard deviation turns out to be close to one, as displayed by the SCREEN block.

```
0.10018E+01
```

See also Blocks STDEVC and STDEVP.

1.151 Block STDEVC

The STDEVC block calculates the standard deviation of its second input signal as long as the condition input remains constant. When this input changes its value, the standard deviation is on output and the successors of the block are executed.



Name	STDEVC
Function	fb0037
Inputs	2 . . . [51]
Outputs	2 . . . [51]
Parameters	0
Strings	0
Group	I

Inputs

1	Condition input c
2	Any signal x_1
n	Any signal x_{n-1}

Outputs

1	Condition input c delayed by one step
2	Standard deviation $\sigma(x_1)$ over all $c =$ const calls
n	Standard deviation $\sigma(x_{n-1})$ over all $c =$ const calls

Parameters

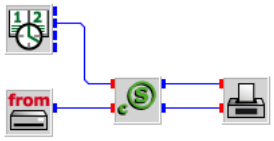
None

Strings

None

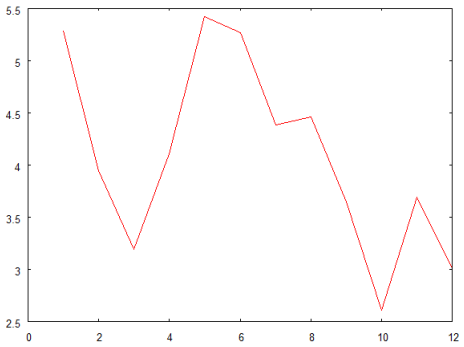
For a description of the calculation of the standard deviation, see block STDEV.

stdevc.vseit



A CLOCK block runs a READ block to read ambient temperature data from file `temperature.dat` located in the `examples/data` directory.

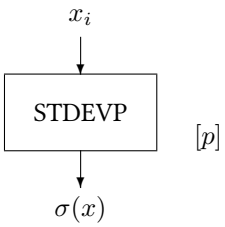
The STDEVC block calculates the monthly values for the standard deviation of the irradiance which varies around 4 degrees Celsius as displayed by the PLOT block.



See also Blocks STDEV and STDEVP.

1.152 Block STDEVP

The STDEVP block calculates the standard deviation of its input signal over a number of steps as defined through its parameter.



Name	STDEVP
Function	fb0020
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	1
Strings	0
Group	I

Inputs

i Any signal x_i

Outputs

i Standard deviation $\sigma(x_i)$ of x_i

Parameters

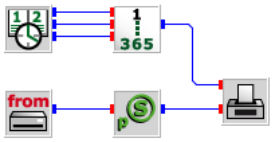
1 Number p of steps to build the standard deviation $\sigma(x_i)$ of x_i

Strings

None

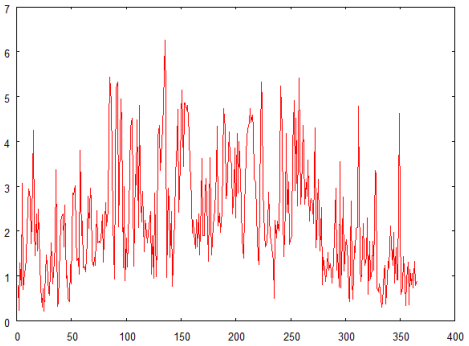
For a description of the calculation of the standard deviation, see block STDEV.

stdevp.vseit



A CLOCK block runs a READ block to read ambient temperature data from file `temperature.dat` located in the `examples/data` directory.

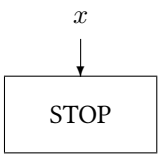
The STDEVP block calculates the daily mean values for the standard deviation of the irradiance which varies between zero an six degrees Celsius as displayed by the PLOT block. The x coordinate is provided by a DOY block (day of the year).



See also Blocks STDEV and STDEVC.

1.153 Block STOP

The STOP block terminates execution of a running INSEL model.



Name	STOP
Function	fb0060
Inputs	1
Outputs	0
Parameters	0
Strings	0
Group	S

Inputs

1 Any input signal x

Outputs

None

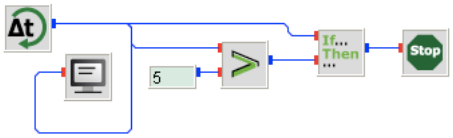
Parameters

None

Strings

None

stop.vseit

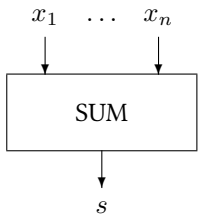


A CYCLEF block opens an endless loop with a sleeping time of one second. The GT block checks whether the output of the CYCLEF block is greater than five. If yes, the STOP block terminates the execution, as can be seen from the SCREEN block's output.

```
0.0000000
1.0000000
2.0000000
3.0000000
4.0000000
5.0000000
```

1.154 Block SUM

The SUM block adds up its inputs.



Name	SUM
Function	fb0002
Inputs	1 . . . [999]
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Any value x_1
2	Any value x_2
n	Any value x_n

Outputs

1	Sum s of all x_i for $i = 1 \ldots n \leq 999$

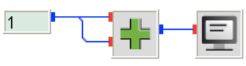
Parameters

None

Strings

None

sum.vseit

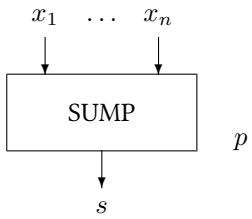


One plus one is

```
2.0
```

1.155 Block SUMP

The SUMP block cumulates its inputs over a given number of calls.



Name	SUMP
Function	fb0041
Inputs	1 . . . [999]
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1	Any value x_1
2	Any value x_2
n	Any value x_n

Outputs

1	Cumulated sum s of all x_i for $i = 1 \ldots n \leq 999$. After p steps the sum s is reset to zero.

Parameters

1	Number $p > 0$ of steps to cumulate the input signals

Strings

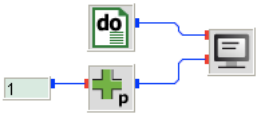
None

Description The block calculates the sum of the inputs $x_1 \ldots x_n$ according to

$$s(t_{m+1}) = s(t_m) + \sum_{i=1}^{n} x_i$$

The sum is reset to zero after p steps.

sump.vseit

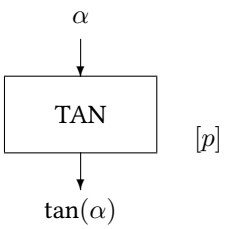


A DO block counts from one to six. The SUMP block cumulates the CONST one and performs a reset every five steps, as specified by the parameter of the SUMP block. The result is shown by a SCREEN block.

```
1.0       1.0
2.0       2.0
3.0       3.0
4.0       4.0
5.0       5.0
6.0       1.0
```

1.156 Block TAN

The TAN block returns the tangent of its input.



Name	TAN
Function	fb0007
Inputs	1
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Any angle α

Outputs

1 Tangent of α

Parameters

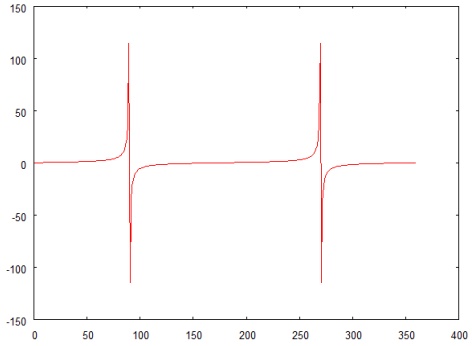
1 Degrees / radians switch p. If p is set to 0 (default) α is interpreted in degrees. If p is set to 1 α is interpreted in radians.

Strings

None

tan.vseit

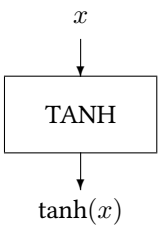
A DO block varies α from 0.5 (in order to avaoid the discontinuities of the tangent function) to 360 degrees in steps of one degree. The TAN block calculates tan(α) and the PLOT block displays the graph on screen.



See also Blocks ACOS, ACOT, ASIN, ATAN, COS, COT, SIN.

1.157 Block TANH

The TANH block returns the hyperbolic tangent of its input.



Name	TANH
Function	fb0068
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any signal x

Outputs

1 Hyperbolic tangent

Parameters

None

Strings

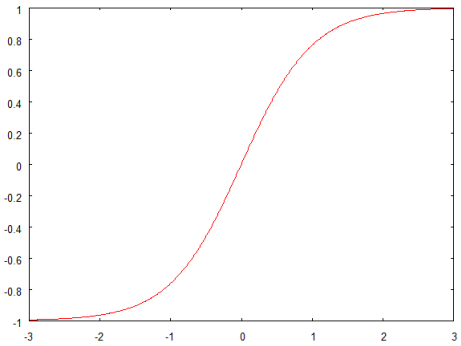
None

Description The hyperbolic tangent is defined as

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

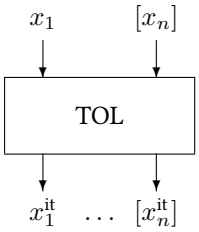
tanh.vseit

A DO block varies its output x from -3 to $+3$ in steps of 0.01. The TANH block calculates $\tanh(x)$ and the PLOT block displays the graph on screen.



1.158 Block TOL

The TOL block defines the iteration signal in a loop set up by a block of group L.



Name	TOL
Function	fb0056
Inputs	1 . . . [50]
Outputs	1 . . . [50]
Parameters	0
Strings	0
Group	-L

Inputs

1 Any signal x_1 from a block of group L
2 Any signal x_2 from a block of group L

Outputs

1 Same value as x_1 but interpreted as iteration signal.
2 Same value as x_2 but interpreted as iteration signal.

Parameters

None

Strings

None

Description

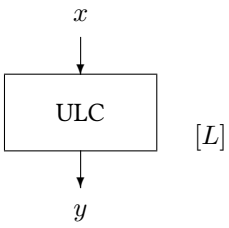
The TOL block (top of loop) is mainly used by the sorting algorithm to reconstruct L-block loops. All blocks using the output signal of the TOL block and contributing to the corresponding L-block's input are sorted into the corresponding iteration process.

At run time, this block performs no operation on the input signal, i. e., $x^{\text{it}} = x$.

The block alone makes no sense without a corresponding L-block. For an example, see block LOOP, MPP, or NULL, for instance.

1.159 Block ULC

The ULC block tests whether the input signal is upward crossing a threshold level.



Name	ULC
Function	fb0039
Inputs	1
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Any input signal x.

Outputs

1 Indicator y for upward level crossing, y is set to one if x has crossed L from $x < L$, else y is set to zero

Parameters

1 Threshold level L (default $L = 0$)

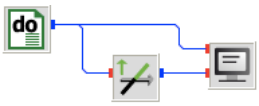
Strings

None

Description The output of this block is defined as

$$y = \begin{cases} 1 & \text{if } x \text{ has crossed } L \text{ from } x \leq L \text{ to } x > L \\ 0 & \text{else} \end{cases}$$

ulc.vseit



A DO block counts from one to five. The output is used by an ULC block with threshold 2.5. Both the counter and the output of the ULC block are displayed using the SCREEN block.

```
1.   0.
2.   0.
3.   1.
4.   0.
5.   0.
```

See also Block DLC.

1.160 Block VECTOR

The VECTOR block defines a constant vector.

Name	VECTOR
Function	fb0079
Inputs	0
Outputs	2
Parameters	2 . . . [1001]
Strings	0
Group	C

Inputs

None

Outputs

None

Parameters

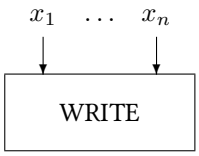
1	Number N of vector elemets
i	x_1 Vector components $i = 1, ...N$

Strings

None

1.161 Block WRITE

The WRITE block writes input data to a file with sequential access.



mode,
fn, format

Name	WRITE
Function	fb0017
Inputs	1 . . . [200]
Outputs	0
Parameters	0 . . . [3]
Strings	0 . . . [3]
Group	S

Inputs

1 Any value x_1 to be written to file fn (Inputs for file name qualifiers are ignored)
2 Any value x_2 to be written to file fn
n Any value x_n to be written to file fn with $n \leq 200$

Outputs

None

Parameters

1 Mode
- 0 If file fn already exists, an error message is given (default)
- 1 If file fn already exists, it is overwritten
- 2 If file fn already exists, new records are appended

2 Suppress file name qualifier inputs (Deprecated! This parameter was basically never used, and can be safely ignored)
- 0 No (default)
- 1 Yes

3 Separator (Deprecated!)
- 0 Blank (default)
- 1 Comma

2 Semicolon

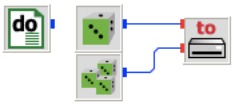
Strings

1 File name fn. If the first character of the file name without path) is a # the name is parsed for variable qualifiers – see Description for further details. If SP(1) is empty a default file `insel.tmp` is generated with in the temporary INSEL directory.
2 Fortran format
3 Optional headline

Remarks JS 8 Dec 2016: Parameters for non-space separators and optional headline added. However, works only for .insel files (.vseit does not support empty string parameters) and implemented only for filenames without qualifiers.

Remarks Integers must be written in format Fx.0, where x is the number of bytes.

write.vseit



A DO block counts from 1 to 8760 (the number of hours of a non-leap year) and two different random number sets are generated by the blocks GASDEV and RAN1. The WRITE block writes these numbers to a data file named `myRandom.dat`.

The first records of the resulting file are these:

```
9.92046744E-02   1.6216065
1.86635051E-02 -0.39489648
0.80818379     -0.33821103
0.75034720      0.53852010
0.79094255     -0.42136794
```

Of course, it is necessary to have write access to the directory where the output file shall be located.

Variable file names If the first character of a specified parameter is a # character the file name is considered variable and is parsed by INSEL. Up to six qualifiers can be used as place holders for digits. The qualifiers have the format `%nx`, where `n` is a positive digit in the range of 1 to 9. `x` is a character out of `YMDhms` – reminding on their most frequent use of date and time information in data file names.

The numerical values for the `n` bytes must be provides as block inputs. `Y` is used with the first input, `M` with the second, and so forth.

Example The file name `#myName%4Y.dat` results in `myName2011.dat`, assuming that the first input has a value of 2011.

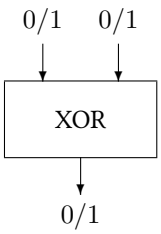
Adding a second input which contains values for the months of a simulation, a qualified file name would be `#myName%4Yplus%2M.dat`. For a simulation running over a complete year the WRITE block creates files with names `myName2011plus1.dat`, `myName2011plus2.dat`, ... `myName2011plus11.dat`, `myName2011plus12.dat`.

If the months from January to September shall be written with leading zeros the qualifier `#myName%4Yplus%2.2M.dat` is accepted and creates files `myName2011plus01.dat`, `myName2011plus02.dat`, and so forth. The same rules apply to all blocks which read data, too.

See also Blocks READ, READD, and READN.

1.162 Block XOR

The XOR block checks whether its first input is equal to one and the second one is equal to zero or (exclusive) the first input is equal to zero and the second one is equal to one.



Name	XOR
Function	fb0030
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Any value x_1 close to one or zero
2 Any value x_2 close to one or zero

Outputs

1 y is set to one if $x_1 \in [0.5, 1.5)$ and $x_2 \in (-0.5, 0.5)$ or vice versa, otherwise y is set to zero

Parameters

None

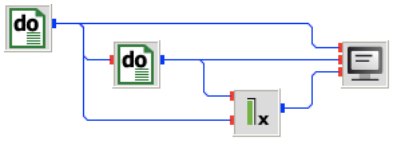
Strings

None

Description The output signal y is calculated from the inputs x_1 and x_2 according to

$$y = \begin{cases} 1 & \text{if } (x_1 \in [0.5, 1.5) \text{ and } x_2 \in (-0.5, 0.5)) \\ & \text{or } (x_1 \in (-0.5, 0.5) \text{ and } x_2 \in [0.5, 1.5)) \\ 0 & \text{else} \end{cases}$$

xor.vseit



Two nested DO blocks deliver all combinations of zero and one, connected to an XOR block. The result is displayed by the SCREEN block.

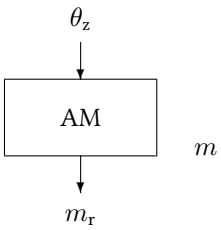
```
0.0   0.0   0.0
0.0   1.0   1.0
1.0   0.0   1.0
1.0   1.0   0.0
```

See also Blocks AND, OR.

2 :: Energy Meteorology

2.1 Block AM

The AM block calculates the relative air mass.



Name	AM
Function	em0025
Inputs	1
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Zenith angle θ_z in degrees

Outputs

1 Relative air mass m_r

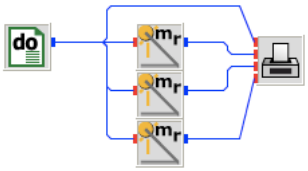
Parameters

1 Model
- 0 Just $\cos \theta_z$
- 1 Kasten 1966
- 2 Kasten 1993

Strings

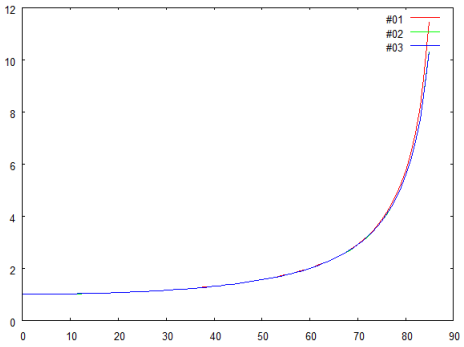
None

am.vseit



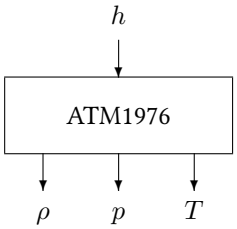
A DO block delivers zenith angles between zero and 85 degrees in steps of one. Three AM blocks calculate the air mass for the three available models.

The PLOT block shows that there is hardly any difference to detect between the three model equations.



2.2 Block ATM1976

The ATM1976 block calculates the properties of the U.S. Standard Atmosphere 1976 to 86 km.



Name	ATM1976
Function	em0024
Inputs	1
Outputs	3
Parameters	0
Strings	0
Group	S

Inputs

1 Geometric altitude h / km. Heights greater than 100 km are limited to 100 km.

Outputs

1 Density / sea-level standard density
2 Pressure / sea-level standard pressure
3 Temperature / sea-level standard temperature

Parameters

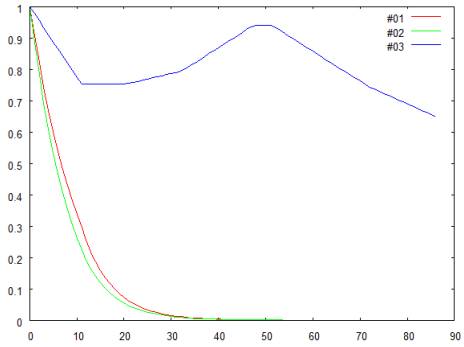
None

Strings

None

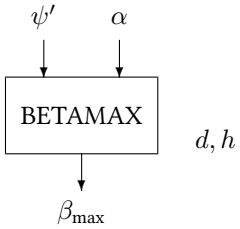
atm1976.vseit

A DO block provides heights from 0 to 86 km in steps of 1 km. The ATM1976 blocks calculates density, pressure and temperature ratios, which are displayed by a PLOT block.



2.3 Block BETAMAX

The BETAMAX block calculates the maximum possible tilt angle such that rows of collector surfaces do not shade each other.



Name	BETAMAX
Function	em0031
Inputs	2
Outputs	1
Parameters	2
Strings	0
Group	S

Inputs

1	Difference between surface azimuth and azimuth of the Sun ψ' / °
2	Solar elevation α / °

Outputs

1	Maximum tilt angle β_{max} / °

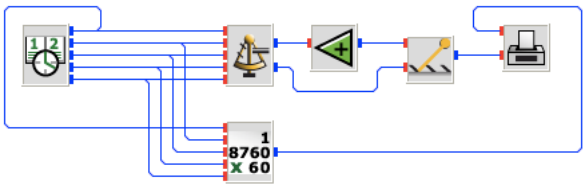
Parameters

1	Distance between module rows d / m
2	Height of modules h / m

Strings

None

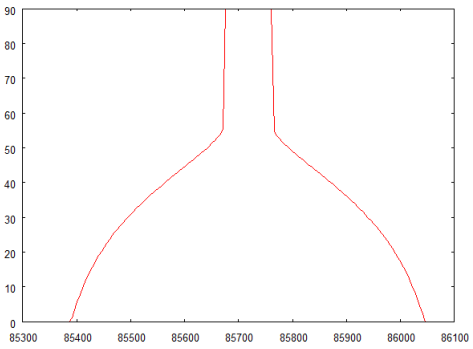
betamax.vseit



A CLOCK block drives this simulation between 06:00 o'clock in the morning until 07:00 p.m. in steps of five minutes on the first of March 2010. The SUNAE block is used to calculate the solar azimuth and elevation. An OFFSET block with parameter -180 is used to calculate the difference between surface azimuth and solar azimut, as required by the BETAMAX block.

The parameters of the BETAMAX block are set to 3.65 meters for the distance between the module rows and 2 meters for the height of the modules.

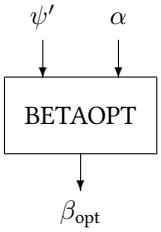
The Gregorian date of the CLOCK block is converted into the minute of the year by a MOY block, which is used as x-coordinate by the PLOT block.



It can be seen from the graph, that around noon no shading is to be expected, regardless of the tilt angle, while the receiver rows have shading problems depending on the actual tilt angle.

2.4 Block BETAOPT

The BETAOPT block calculates the optimum tilt angle for a one-axis tracked receiver with horizontal axis.



Name	BETAOPT
Function	em0032
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Difference between surface azimuth and azimuth of the Sun / °
2 Solar elevation / °

Outputs

1 Optimum tilt angle / °

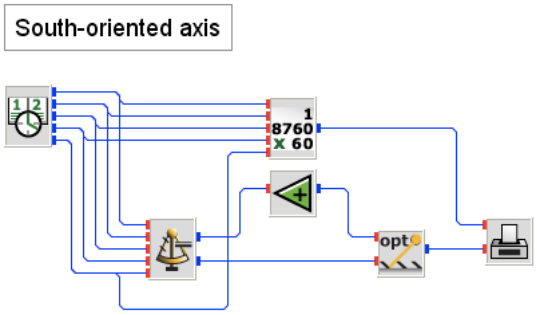
Parameters

None

Strings

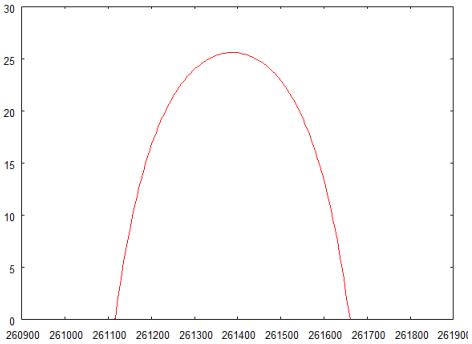
None

betaopt.vseit



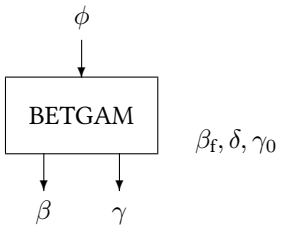
A CLOCK block drives this simulation between 05:00 o'clock in the morning until 09:00 p.m. in steps of five minutes on the first of July 2010. The SUNAE block is used to calculate the solar azimuth and elevation. An OFFSET block with parameter -180 is used to calculate the difference between surface azimuth and solar azimuth, as required by the BETAOPT block.

The Gregorian date of the CLOCK block is converted into the minute of the year by a MOY block, which is used as x-coordinate by the PLOT block.



2.5 Block BETGAM

The BETGAM block calculates the effective tilt and azimuth angle of a tilted, one-axis tracked receiver area.



Name	BETGAM
Function	em0037
Inputs	1
Outputs	2
Parameters	1 . . . [3]
Strings	0
Group	S

Inputs

1 Rotation angle $\phi \in [-90°, 90°]$, east negative, west positive

Outputs

1 Effective tilt angle β / °
2 Effective azimuth angle γ / °

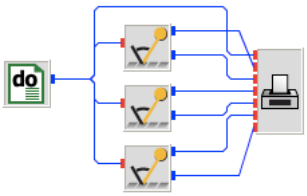
Parameters

1 Tilt angle of fixed rotation axis β_f / °
2 Deviation of axis from north/south direction (clockwise positive) (optional) δ / ° $\in [-90°, 90°]$
3 Azimuth angle if undefined γ_0 / ° (optional)

Strings

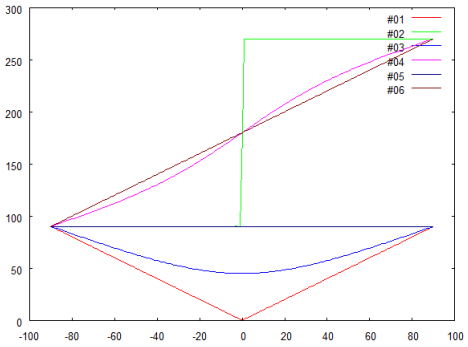
None

betgam.vseit



A DO block varies a rotation axis from −90 to +90 degrees in steps of one degree. The rotation axis is connected to three BETGAM blocks with different orientations: The top block uses a horizontal rotation axis, the middle block a tilt angle of the rotation axis of 45 degrees, the lower block simulates a vertical rotation axis. In all three cases the deviation from the north-south direction is zero.

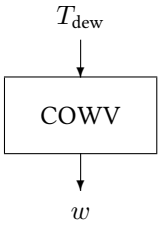
The PLOT block shows the effective tilt and effective azimuth angles.



As could be expected, the horizontal axis faces east in the morning and west in the afternoon (green curve). The tilt angle of the horizontal case is also easy to interpret: 90 degrees at −90 degrees, linear down to zero at true solar noon and linear up to 90 degrees.

2.6 Block COWV

The COWV block calculates content of water vapour in the atmosphere as a function of the dewpoint temperature.



Model

Name	COWV
Function	em0035
Inputs	1
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Dewpoint temperature T_{dew} / ° C

Outputs

1 Content of water vapour w / cm

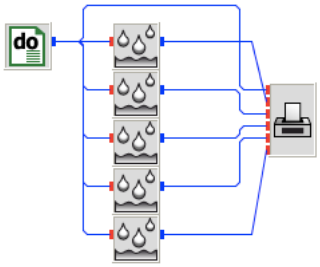
Parameters

1 Model
- 0 Bolsegna (Hourly means)
- 1 Wright et al. (Three-hour means)
- 2 Reitan (Monthly means)
- 3 Atwater et al. (Monthly means)
- 4 Smith (Annual means)

Strings

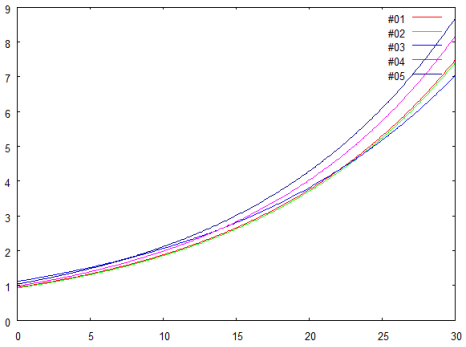
None

cowv.vseit



A DO block varies dewpoint temperature data from zero to a 30 degrees Celsius in steps of 0.1. The dewpoint temperature is connected to five COWV blocks with the five different model equations starting from Bolsegna (upper block) to Smith (lowest block).

The PLOT block shows the respective water vapor contents.



2.7 Block DINCLI

The DINCLI block gives monthly standard values for Germany. Only input is the number of the month between 1 and 12.

Name	DINCLI
Function	em0039
Inputs	1
Outputs	6
Parameters	0
Strings	0
Group	S

Inputs

1 Month

Outputs

1 Standard temperature for Germany / °C
2 Standard radiation on north surface for Germany / W /m^{-2}
3 Standard radiation on east surface for Germany / W /m^{-2}
4 Standard radiation on south surface for Germany / W /m^{-2}
5 Standard radiation on west surface for Germany / W /m^{-2}
6 Standard radiation on horizontal surface for Germany / W /m^{-2}

Parameters

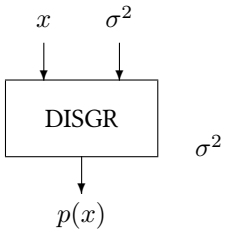
None

Strings

None

2.8 Block DISGR

The DISGR block calculates the Gordon Reddy distribution.



Name	DISGR
Function	em0023
Inputs	1 . . . [2]
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1	Independent variable x
2	Variance $\sigma^2(x)$

Outputs

1	Probability density $p(x)$

Parameters

1	Variance $\sigma^2(x)$

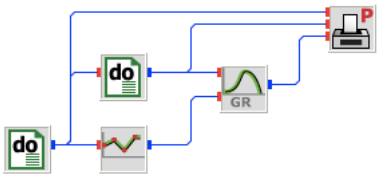
Strings

None

Remarks

If a second input is connected to a DISGR block, the value of the block parameter σ^2 is ignored.

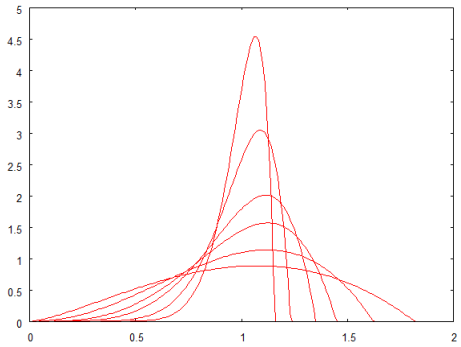
disgr.vseit



Two nested DO blocks count from one to six (outer block), and vary the Gordon Reddy variable from zero to two in steps of 0.01 (inner block), respectively. The POLYG block delivers different variance values due the parameter setting

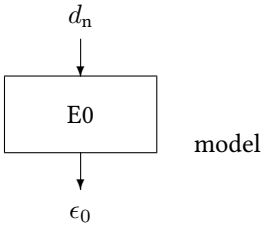
```
6
1 0.01
2 0.02
3 0.04
4 0.06
5 0.1
6 0.15
```

The PLOTP block shows the probability density functions for the different variances.



2.9 Block E0

The E0 block calculates the eccentricity correction of the Earth's orbit.



model

Name	E0
Function	em0002
Inputs	1
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Day of the year d_n

Outputs

1 Eccentricity ϵ_0

Parameters

1 Approximation
- 0 Spencer
- 1 Duffie Beckman

Strings

None

Description

The path of the Earth's orbit is not exactly a circle but an ellipse with a small eccentricity and with the Sun in one of the foci. The real distance r_{se} between Sun and Earth varies between 0.983 AU $= 147.1$ Gm (perigee) and 1.017 AU $= 152.1$ Gm (apogee).

The eccentricity correction factor of the Earth's orbit defined as $\epsilon_0 = (\bar{r}_{se}/r_{se})^2$ can be approximated by

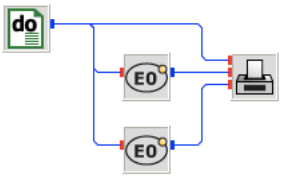
$$\epsilon_0 = 1 + 0.033 \cdot \cos(2\pi d_n/365)$$

where $d_n \in [1, 365]$ denotes the day number of the year. The Fourier series as given by Spencer

$$\begin{aligned}\epsilon_0 = {} & 1.00010 + 0.034221 \cdot \cos d + 0.001280 \cdot \sin d \\ & + \ 0.000719 \cdot \cos 2d + 0.000077 \cdot \sin 2d\end{aligned}$$

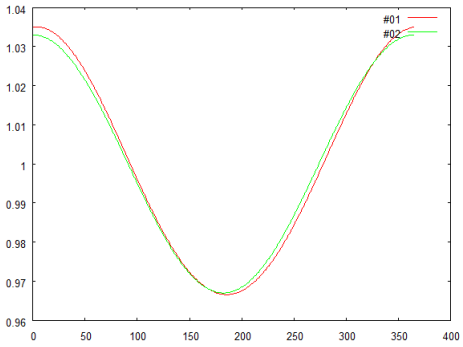
is more accurate.

e0.vseit



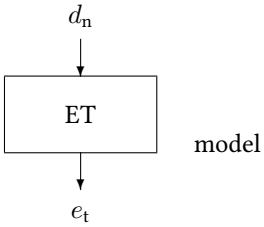
A DO block counts through the days of a year from 1 to 365. The two E0 blocks calculate the eccentricity after Spencer (upper block) and Duffie and Beckman (lower block).

The PLOT block shows the result.



2.10 Block ET

The ET block calculates the equation of time.



Name	ET
Function	em0003
Inputs	1
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Day of the year d_n

Outputs

1 Equation of time e_t / min

Parameters

1 Approximation
- 0 Spencer
- 1 Wagner
- 2 Duffie Beckman
- 3 DIN 5034 Teil 2
- 4 Woolf 1968
- 5 Lamm 1981

Strings

None

Description

The equation of time e_t describes the difference between true solar time t_τ and mean solar time t_μ

$$e_t = t_\tau - t_\mu$$

A relatively simple formula for the equation of time is given by Duffie and Beckman

$$e_t = 9.87 \sin(2B) - 7.53 \cos(B) - 1.5 \sin(B)$$

where

$$B = \frac{360(d_n - 81)}{364}$$

According to Spencer the Fourier series

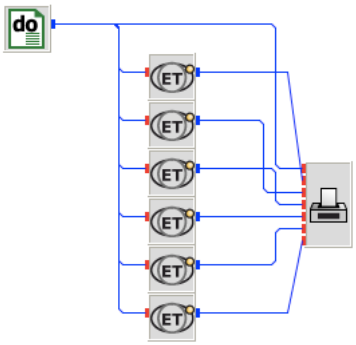
$$\begin{aligned} e_t \quad = \quad & (0.000075 + 0.001868 \cos(d) - 0.032077 \sin(d) \\ & -0.014615 \cos(2d) - 0.04089 \sin(2d))(180 \cdot 4/\pi) \end{aligned}$$

yields more precise results. d denotes the day angle defined by

$$d = \frac{2\pi(d_n - 1)}{365}$$

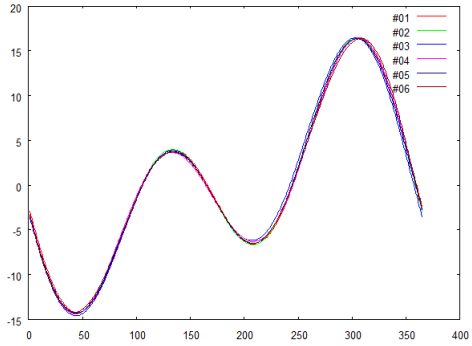
Both formulas are implemented in the ET block.

et.vseit



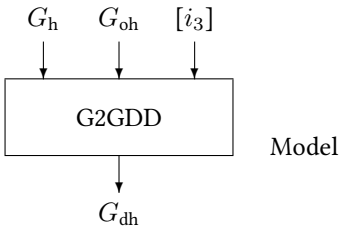
A DO block provides the days of a year starting from 1 to 365. Six ET blocks calculate the equation of time for the different approximations starting with Spencer's Fourier series (upper block) to Lamm's approximation (lower block).

The PLOT block shows that the differences are nearly negligible.



2.11 Block G2GDD

The G2GDD block returns the diffuse radiation according to various correlations for daily data.



Name	G2GDD
Function	em0010
Inputs	2 . . . [3]
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Global radiation G_h / W m^{-2} on a horizontal plane
2 Global radiation outside the atmosphere G_{oh} / W m^{-2} on a horizontal plane
3 Model dependent (see Parameters section for further details)

Outputs

1 Diffuse radiation G_{dh} / W m^{-2} on a horizontal plane

Parameters

1 Correlation
- 0 Liu, Jordan
- 1 Ruth, Chant
- 2 Collares-Pereira
- 3 Vignola, McDaniels
- 4 Erbs, Klein, Duffie – this correlation requires the sunrise hour angle ω_{sr} / ° as third block input.
- 5 Vignola, McDaniels – this correlation requires the day of the year DOY as third block input.

Strings

None

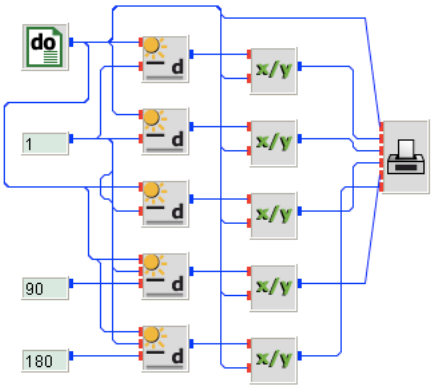
Description

The block calculates the fraction of daily diffuse and global radiation on a horizontal surface mainly on the basis of the clearness index which is defined as

$$k_t = \frac{G_h}{G_{oh}}$$

Through the setting of the first block parameter one of several correlations can be chosen.

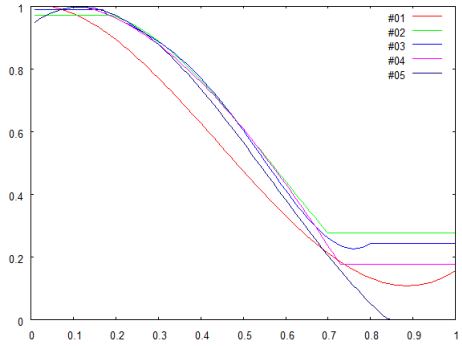
g2gdd.vseit



A DO block is used to vary the (normalised) global radiation on a horizontal surface from 0.01 to 1 in steps of 0.01. The extraterrestrial radiation is set to one by a CONST block. Hence the two inputs to the G2GDD blocks correspond exactly to a clearness index. Usually, the second input to the G2GDD block comes from block GOH.

The DIV blocks calculate the diffuse fraction by dividing the diffuse radiation from the G2GDD blocks by the output of the DO block.

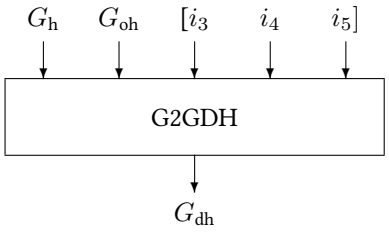
As shown by the PLOT block the results for all correlations are quite similar.



See also Block G2GDH and G2GDM

2.12 Block G2GDH

The G2GDH block returns the diffuse radiation according to various correlations for radiation hourly data.



Model

Name	G2GDH
Function	em0009
Inputs	2 . . . [5]
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Global radiation G_h / W m^{-2} on a horizontal plane
2 Global radiation outside the atmosphere G_{oh} / W m^{-2} on a horizontal plane
3..5 Model dependent (see Parameters section for further details)

Outputs

1 Diffuse radiation G_{dh} / W m^{-2} on a horizontal plane

Parameters

1 Correlation
- 0 Orgill, Hollands
- 1 Erbs, Klein, Duffie
- 2 Hollands
- 3 Reindl, Beckman, Duffie
- 4 Hollands, Chra – this correlation requires the ground albedo ρ as third block input.
- 5 Skartveit, Olseth – this correlation requires the zenith angle θ_z / ° as third block input.
- 6 Reindl, Beckman, Duffie – this correlation requires the zenith angle θ_z / ° as third block input.

7 Reindl, Beckman, Duffie – this correlation requires three additional inputs: the zenith angle θ_z / ° as third block input, the ambient temperature T_a / °C as fourth input and the relative humidity as input number five.

Strings

None

Description

The block calculates the fraction of hourly diffuse and global radiation on a horizontal surface mainly on the basis of the clearness index which is defined as

$$k_t = \frac{G_h}{G_{oh}}$$

Through the setting of the first block parameter one of several correlations can be chosen.

Orgill/Hollands

The the Orgill/Hollands correlation is given by

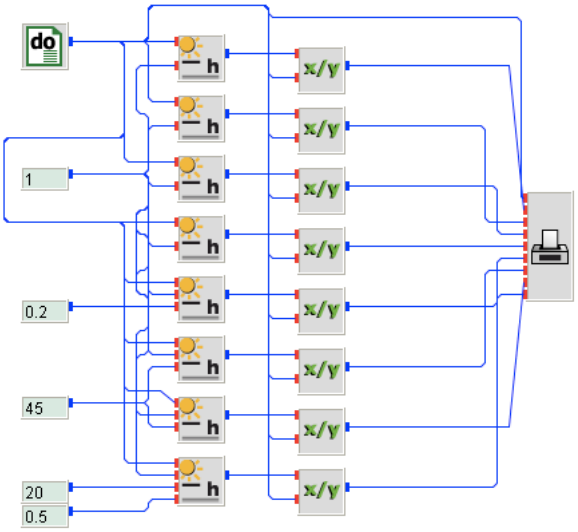
$$\frac{G_{dh}}{G_h} = \begin{cases} 1.0 - 0.249\,k_t & \text{if } 0 \le k_t \le 0.35 \\ 1.557 - 1.84\,k_t & \text{if } 0.35 < k_t \le 0.75 \\ 0.177 & \text{if } k_t > 0.75 \end{cases}$$

Erbs/Klein/ Duffie

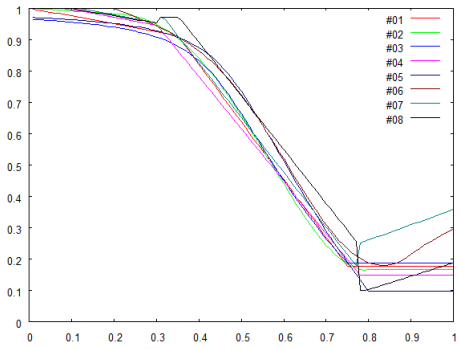
The Erbs/Klein/Duffie-Korrelation correlation is given by

$$\frac{G_{dh}}{G_h} = \begin{cases} 1.0 - 0.09\,k_t & \text{if } 0 \le k_t \le 0.22 \\ 0.9511 - 0.1604\,k_t + 4.388\,{k_t}^2 & \\ \quad -16.638\,{k_t}^3 + 12.336\,{k_t}^4 & \text{if } 0.22 < k_t \le 0.80 \\ 0.165 & \text{if } k_t > 0.80 \end{cases}$$

g2gdh.vseit



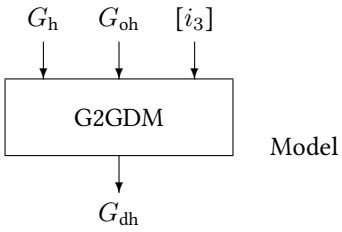
The idea of this example is exactly the same as in the example for block G2GDD only that correlations for hourly data are presented here. For further details, please refer to the example of block G2GDD.



See also Block G2GDD and G2GDM

2.13 Block G2GDM

The G2GDM block returns the diffuse radiation according to various correlations for monthly data.



Name	G2GDM
Function	em0011
Inputs	2 . . . [3]
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Global radiation G_h / W m^{-2} on a horizontal plane
2 Global radiation outside the atmosphere G_{oh} / W m^{-2} on a horizontal plane
3 Model dependent (see Parameters section for further details)

Outputs

1 Diffuse radiation G_{dh} / W m^{-2} on a horizontal plane

Parameters

1 Correlation
- 0 Liu, Jordan
- 1 Erbs, Klein, Duffie
- 2 Vignola, McDaniels
- 3 Collares-Pereira – this correlation requires the sunrise hour angle ω_{sr} / ° as third block input.
- 4 Erbs, Klein, Duffie – this correlation requires the sunrise hour angle ω_{sr} / ° as third block input.
- 5 Vignola, McDaniels – this correlation requires the day of the year DOY as third block input.

Strings

None

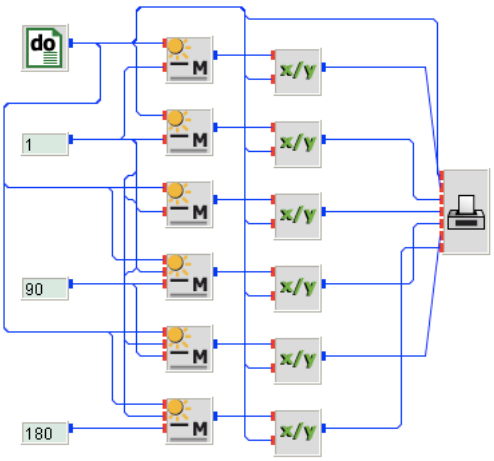
Description

The block calculates the fraction of monthly diffuse and global radiation on a horizontal surface mainly on the basis of the clearness index which is defined as

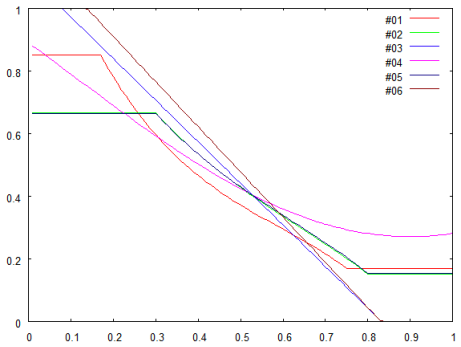
$$k_t = \frac{G_h}{G_{oh}}$$

Through the setting of the first block parameter one of several correlations can be chosen.

g2gdm.vseit



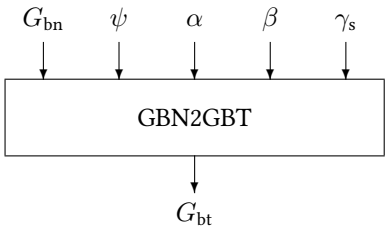
The idea of this example is exactly the same as in the example for block G2GDD only that correlations for monthly data are presented here. For further details, please refer to the example of block G2GDD.



See also Block G2GDD and G2GDH

2.14 Block GBN2GBT

The GBN2GBT block converts beam radiation normal to tilted.



Name	GBN2GBT
Function	em0019
Inputs	5
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Beam radiation normal G_{bn}/ W m^{-2}
2 Azimuth of the Sun ψ / °
3 Elevation of the Sun α / °
4 Surface tilt angle β / °
5 Surface azimuth γ_s / °

Outputs

1 Beam radiation tilted G_{bt} / W m^{-2}

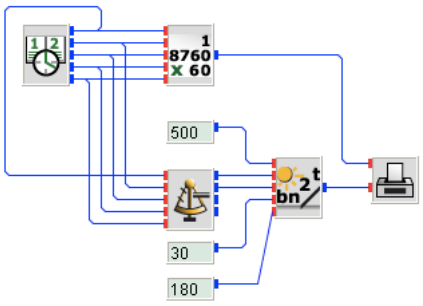
Parameters

None

Strings

None

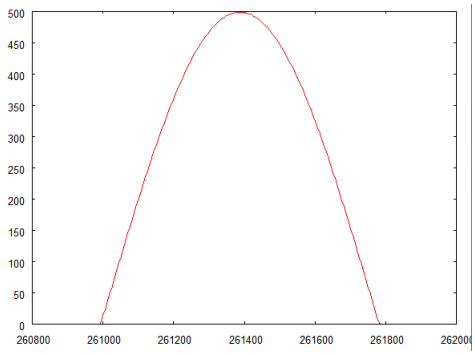
gbn2gbt.vseit



A CLOCK block simulates the first of June 2010 between 04:00 o'clock in the morning until 10:00 p.m., i. e., 22:00 hours in steps of five minutes. Assuming a constant light source with 500 Watt per square meter beam irradiance from the direction of the Sun, the GBN2GBT block calculates how much of this beam radiation arrives at a suface facing the south (azimuth 180 degrees, in INSEL convention) with a tilt angle of 30 degrees – as defined by two CONST blocks.

The SUNAE block calculates solar azimuth and elevation after Spencer's model for the coordinates of Stuttgart, Germany (latitude 48.77 degrees north, longitude 9.18 degrees east, i. e., -9.18 degrees in INSEL convention, time zone 23, i. e., Central European Time). Solar azimuth and elevation are connected to the GBN2GBT block.

A minute of the year block MOY is used as x-coordinate by the PLOT block which displays the result.



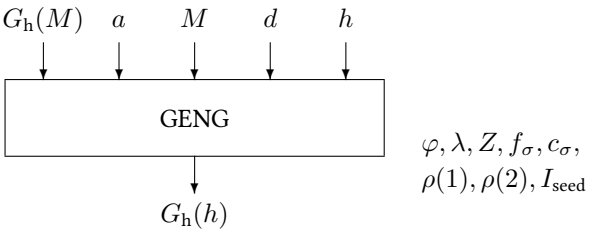
See also Block GH2GT

2.15 Block GENG

Warning: The explicit usage of this block is deprecated, because it uses a west-positive convention for longitudes.

Please use GENG2 instead, with east-positive longitudes. GENG is still used internally by GENG2.

The GENG block generates a series of hourly global radiation data from monthly mean values.



Name	GENG
Function	em0015
Inputs	5 . . . [6]
Outputs	1
Parameters	8
Strings	0
Group	S

Inputs

1 Monthly mean value $G_h(M)$ / W m^{-2} of global radiation on a horizontal plane
2 Year a
3 Month $M \in [1, 12]$
4 Day $d \in [1, 31]$
5 Hour $h \in [0, 23]$
6 Minute $m \in [0, 59]$

Outputs

1 Hourly mean value $G_h(h)$ / W m^{-2} of global radiation on a horizontal plane. When input six is connected, GENG performs a linear interpolation between the two corresponding hours.

Parameters

1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2 Longitude $\lambda \in [0°, 360°)$, west of Greenwich; values east of Greenwich may be used with a minus sign
3 Time zone $Z \in [0, 23]$, Greenwich Mean Time $Z = 0$, Central European Time $Z = 23$.
4 Variance factor f_σ to the Gordon / Reddy correlation, eq **??**; if unknown $f_\sigma = 1$ is recommended
5 Coefficient c_σ corresponding to the year-to-year variability due to different climatic conditions. When c_σ is set to zero the year-to-year variability is omitted. $c_\sigma = 0.185$ approximates North American variability, while $c_\sigma = 0.3$ approximates European variability
6 Autocorrelation coefficient $\rho(1)$ at a lag of one day; if unknown $\rho(1) = 0.3$ is recommended
7 Autocorrelation coefficient $\rho(2)$ at a lag of two days; if unknown $\rho(2) = 0.57\rho(1)$ is recommended
8 Initialization I_{seed} of random number generator

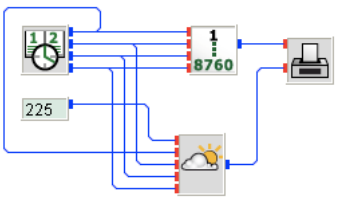
Strings

None

Remarks

The GENG block is basically a convenient combination of the GENGD block and the GENGH block.

geng.vseit



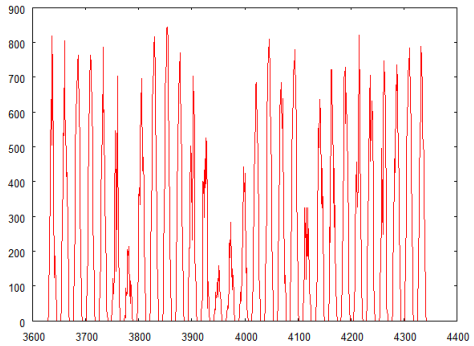
A CLOCK block simulates June 2010 in steps of one hour. The monthly mean irradiance is assumed to be 225 W/m^2 as provided by a CONST block.

In this example, the GENG block generates hourly irradiance data for the location of Stuttgart, Germany.

The Gordon-Reddy variance factor is set to one, the year-to-year variability is set to zero, such that the time series of generated irradiance data represents the monthly mean value as close as possible.

The autocorrelation coefficients are set to their defaults for European climate, i. e., 0.3 and 0.171, respectively. The random number generator is initialized by 4712.

An hour of the year block HOY is used as x-coordinate by the PLOT block which displays the generated irradiance data time series.

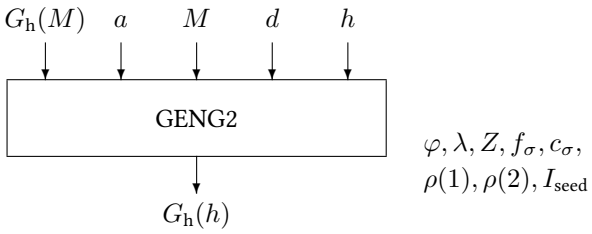


See also Blocks DISGR, GENGD and GENGH.

2.16 Block GENG2

The GENG2 block generates a series of hourly global radiation data from monthly mean values.

Internally, it simply calls GENG which uses an old convention for timezone and longitude.



Name	GENG2
Function	em0115
Inputs	5 . . . [6]
Outputs	1
Parameters	8
Strings	0
Group	S

Inputs

1 Monthly mean value $G_\mathrm{h}(M)$ / W m^{-2} of global radiation on a horizontal plane
2 Year a
3 Month $M \in [1, 12]$
4 Day $d \in [1, 31]$
5 Hour $h \in [0, 23]$
6 Minute $m \in [0, 59]$

Outputs

1 Hourly mean value $G_\mathrm{h}(h)$ / W m^{-2} of global radiation on a horizontal plane. When input six is connected, GENG2 performs a linear interpolation between the two corresponding hours.

Parameters

1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2 Longitude $\lambda \in [-180°, 180°]$, **east of Greenwich**; values west of Greenwich may be used with a minus sign

3	UTC Time zone $Z \in [-12, 12]$, Greenwich Mean Time $Z = 0$, Central European Time $Z = +1$.
4	Variance factor f_σ to the Gordon / Reddy correlation, eq **??**; if unknown $f_\sigma = 1$ is recommended
5	Coefficient c_σ corresponding to the year-to-year variability due to different climatic conditions. When c_σ is set to zero the year-to-year variability is omitted. $c_\sigma = 0.185$ approximates North American variability, while $c_\sigma = 0.3$ approximates European variability
6	Autocorrelation coefficient $\rho(1)$ at a lag of one day; if unknown $\rho(1) = 0.3$ is recommended
7	Autocorrelation coefficient $\rho(2)$ at a lag of two days; if unknown $\rho(2) = 0.57\rho(1)$ is recommended
8	Initialization I_{seed} of random number generator

Strings

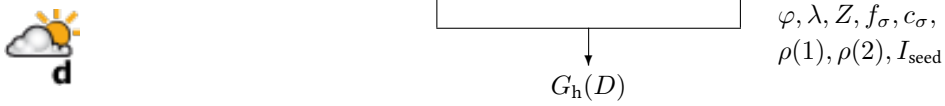
None

2.17 Block GENGD

Warning: The explicit usage of this block is deprecated, because it uses a west-positive convention for longitudes.

Please use GENGD2 instead, with east-positive longitudes. GENGD is still used internally by GENGD2.

The GENGD block generates a series of daily global radiation data from monthly mean values.



Name	GENGD
Function	em0016
Inputs	4
Outputs	1
Parameters	9
Strings	0
Group	S

Inputs

1 Monthly mean value $G_\mathrm{h}(M)$ / W m^{-2} of global radiation on a horizontal plane
2 Year a
3 Month $M \in [1, 12]$
4 Day $d \in [1, 31]$

Outputs

1 Daily mean value $G_\mathrm{h}(d)$ / W m^{-2} of global radiation on a horizontal plane

Parameters

1 Model
  0 Gordon Reddy model
  1 Aguiar Collares-Pereira model
2 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive

3 Longitude $\lambda \in [0°, 360°)$, west of Greenwich; values east of Greenwich may be used with a minus sign

4 Time zone $Z \in [0, 23]$, Greenwich Mean Time $Z = 0$, Central European Time $Z = 23$.

5 Variance factor f_σ to the Gordon / Reddy correlation; if unknown $f_\sigma = 1$ is recommended

6 Coefficient c_σ corresponding to the year-to-year variability due to different climatic conditions. When c_σ is set to zero the year-to-year variability is omitted. $c_\sigma = 0.185$ approximates North American variability, while $c_\sigma = 0.3$ approximates European variability

7 Autocorrelation coefficient $\rho(1)$ at a lag of one day; if unknown $\rho(1) = 0.3$ is recommended

8 Autocorrelation coefficient $\rho(2)$ at a lag of two days; if unknown $\rho(2) = 0.57\rho(1)$ is recommended

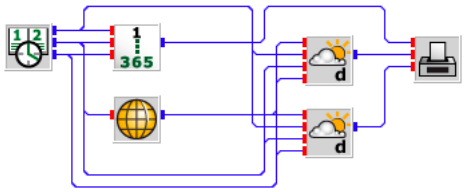
9 Initialization I_{seed} of random number generator

Strings

None

---

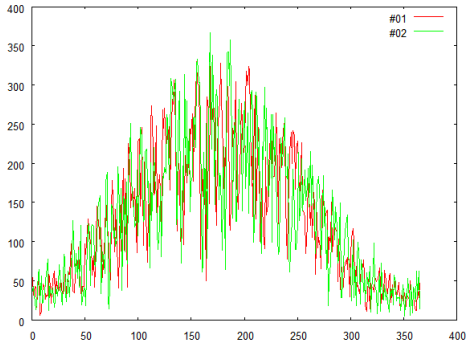
gengd.vseit



A CLOCK runs through the year 2010 in steps of one day. Monthly mean irradiance data are read from the INSEL weather data base by the MTM block for the location of Stuttgart, Germany.

Two GENGD blocks generate daily irradiance data, the upper block uses the Gordon and Reddy model, the lower one the Markov model approach proposed by Aguilar and Collares-Pereira.

A day of the year block DOY is used as x-coordinate by the PLOT block which displays the two generated irradiance data time series.

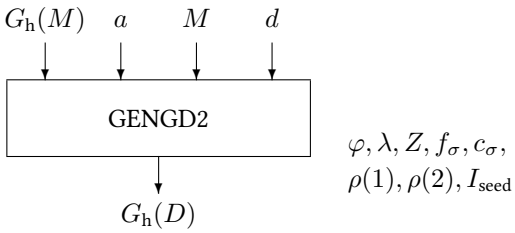


See also Blocks DISGR, GENG and GENGH.

2.18 Block GENGD2

The GENGD2 block generates a series of daily global radiation data from monthly mean values.

Internally, it simply calls GENGD which uses an old convention for timezone and longitude.



Name	GENGD2
Function	em0116
Inputs	4
Outputs	1
Parameters	9
Strings	0
Group	S

Inputs

1 Monthly mean value $G_h(M)$ / W m^{-2} of global radiation on a horizontal plane
2 Year a
3 Month $M \in [1, 12]$
4 Day $d \in [1, 31]$

Outputs

1 Daily mean value $G_h(d)$ / W m^{-2} of global radiation on a horizontal plane

Parameters

1 Model
  0 Gordon Reddy model
  1 Aguiar Collares-Pereira model
2 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
3 Longitude $\lambda \in [-180°, 180°)$, **east of Greenwich**; values west of Greenwich may be used with a minus sign

4	UTC Time zone $Z \in [-12, 12]$, Greenwich Mean Time $Z = 0$, Central European Time $Z = +1$.
5	Variance factor f_σ to the Gordon / Reddy correlation; if unknown $f_\sigma = 1$ is recommended
6	Coefficient c_σ corresponding to the year-to-year variability due to different climatic conditions. When c_σ is set to zero the year-to-year variability is omitted. $c_\sigma = 0.185$ approximates North American variability, while $c_\sigma = 0.3$ approximates European variability
7	Autocorrelation coefficient $\rho(1)$ at a lag of one day; if unknown $\rho(1) = 0.3$ is recommended
8	Autocorrelation coefficient $\rho(2)$ at a lag of two days; if unknown $\rho(2) = 0.57\rho(1)$ is recommended
9	Initialization I_{seed} of random number generator

Strings

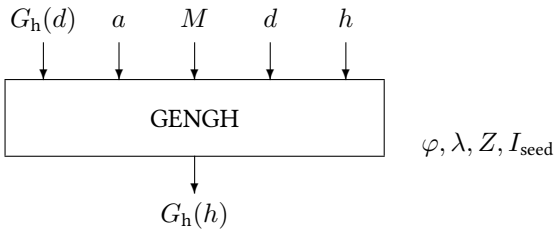
None

2.19 Block GENGH

Warning: The explicit usage of this block is deprecated, because it uses a west-positive convention for longitudes.

Please use GENGH2 instead, with east-positive longitudes. GENGH is still used internally by GENGH2.

The GENGH block generates a series of hourly global radiation data from daily mean values.



Name	GENGH
Function	em0017
Inputs	5 . . . [6]
Outputs	1
Parameters	4
Strings	0
Group	S

Inputs

1	Daily mean value $G_h(d)$ / W m^{-2} of global radiation on a horizontal plane
2	Year a
3	Month $M \in [1, 12]$
4	Day $d \in [1, 31]$
5	Hour $h \in [0, 23]$
6	Minute $m \in [0, 59]$

Outputs

1	Hourly mean value $G_h(h)$ / W m^{-2} of global radiation on a horizontal plane.When input six is connected, GENGH performs a linear interpolation between the two corresponding hours.

Parameters

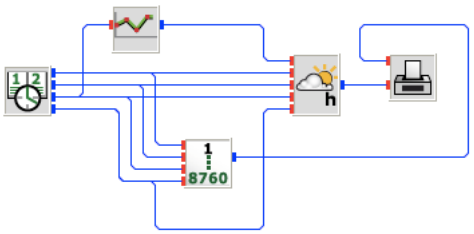
1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2 Longitude $\lambda \in [0°, 360°)$, west of Greenwich; values east of Greenwich may be used with a minus sign
3 Time zone $Z \in [0, 23]$, Greenwich Mean Time $Z = 0$, Central European Time $Z = 23$.
4 Initialization I_{seed} of random number generator

Strings

None

---

gengh.vseit

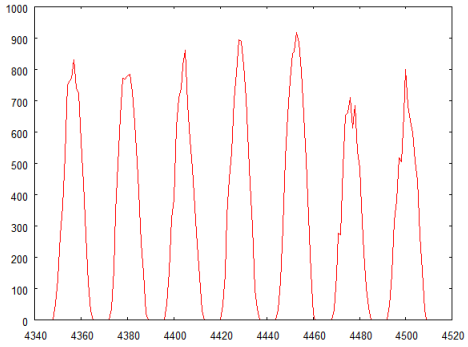


A CLOCK block runs through the first week of July 2010 in steps of one hour and a POLYG block with the parameters

```
7
1.00    301.03
2.00    302.37
3.00    295.50
4.00    321.24
5.00    344.02
6.00    254.34
7.00    272.85
```

provides daily means of global radiation data, which are expanded by the GENGH block to hourly values for the location of Stuttgart, Germany.

An hour of the year block HOY is used as x-coordinate by the PLOT block which displays the generated irradiance data time series.

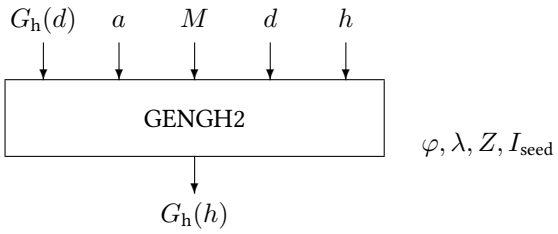


See also Blocks GENG and GENGD.

2.20 Block GENGH2

The GENGH2 block generates a series of hourly global radiation data from daily mean values.

Internally, it simply calls GENGH which uses an old convention for timezone and longitude.



Name	GENGH2
Function	em0117
Inputs	5 . . . [6]
Outputs	1
Parameters	4
Strings	0
Group	S

Inputs

1 Daily mean value $G_h(d)$ / W m^{-2} of global radiation on a horizontal plane
2 Year a
3 Month $M \in [1, 12]$
4 Day $d \in [1, 31]$
5 Hour $h \in [0, 23]$
6 Minute $m \in [0, 59]$

Outputs

1 Hourly mean value $G_h(h)$ / W m^{-2} of global radiation on a horizontal plane.When input six is connected, GENGH2 performs a linear interpolation between the two corresponding hours.

Parameters

1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2 Longitude $\lambda \in [-180°, 180°)$, **east of Greenwich**; values west of Greenwich may be used with a minus sign

3 UTC Time zone $Z \in [-12, 12]$, Greenwich Mean Time $Z = 0$, Central European Time $Z = +1$.

4 Initialization I_{seed} of random number generator

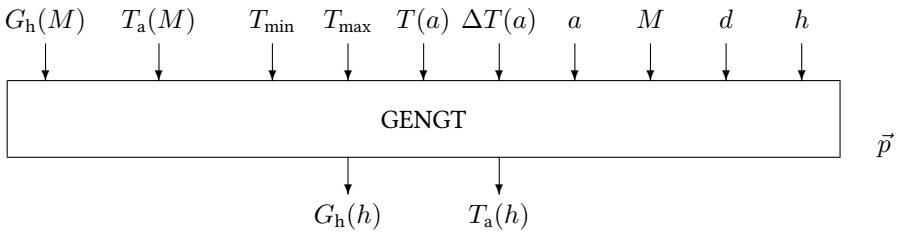
Strings

None

2.21 Block GENGT

Warning: This block is deprecated, because it used a west-positive convention for longitudes. Please use GENGT2 instead, with east-positive longitudes.

The GENGT block generates a series of hourly radiation and temperature data from monthly mean values. Optionally, hourly values of relative humidity may be generated. When a minute input is connected and different from zero, GENGT performs a linear interpolation between the two corresponding hours.



Name	GENGT
Function	em0022
Inputs	10 . . . [12]
Outputs	3
Parameters	8 . . . [10]
Strings	0
Group	S

Inputs

1 Global horizontal irradiance $G_\mathrm{h}(M)$ / W m^{-2} (monthly)
2 Ambient temperature $T_\mathrm{a}(M)$ / °C (monthly)
3 Daily minimum ambient temperature T_min / °C
4 Daily maximum ambient temperature T_max / °C
5 Yearly average ambient temperature $T(a)$ / °C
6 Maximum ambient temperature difference $\Delta T(a)$ / °C
7 Year a
8 Month $M \in [1, 12]$
9 Day $d \in [1, 31]$
10 Hour $h \in [0, 23]$
11 Relative humidity $\varphi(M)$ (monthly)
12 Minute $m \in [0, 59]$

Outputs

1 Global radiation $G_\mathrm{h}(h)$ / W m^{-2}
2 Ambient temperature $T_\mathrm{a}(h)$ / °C
3 Relative humidity $\varphi_(h)$

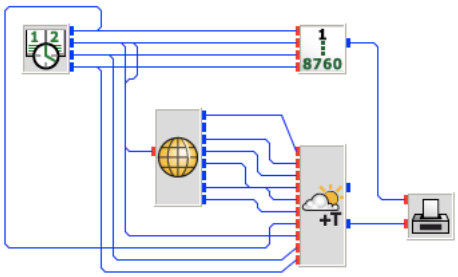
Parameters

1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2 Longitude $\lambda \in [0°, 360°)$, **west of Greenwich**; values east of Greenwich may be used with a minus sign
3 Time zone $Z \in [0, 23]$, Greenwich Mean Time $Z = 0$, Central European Time $Z = 23$.
4 Variance factor f_σ to the Gordon / Reddy correlation, eq **??**; if unknown $f_\sigma = 1$ is recommended
5 Coefficient c_σ corresponding to the year-to-year variability due to different climatic conditions. When c_σ is set to zero the year-to-year variability is omitted. $c_\sigma = 0.185$ approximates North American variability, while $c_\sigma = 0.3$ approximates European variability
6 Autocorrelation coefficient $\rho(1)$ at a lag of one day; if unknown $\rho(1) = 0.3$ is recommended
7 Autocorrelation coefficient $\rho(2)$ at a lag of two days; if unknown $\rho(2) = 0.57\rho(1)$ is recommended
8 Initialization I_seed of random number generator
9 Maximum allowed deviation between given monthly mean temperature and calculated value (0.1°C by default)
10 Maximum number of allowed iterations for temperature synthesis (2000 by default)

Strings

None

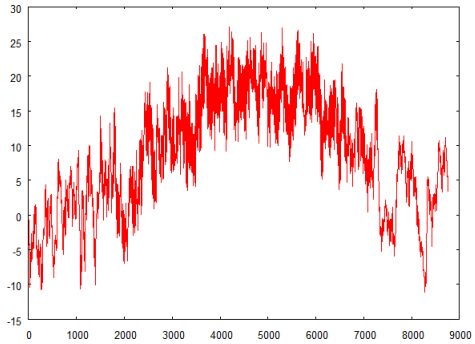
gengt.vseit



A CLOCK block simulates the year 2010 in steps of one hour. An hour of the year block HOY is used to convert the CLOCK block output to a convenient x-coordinate for a PLOT block.

The input data required by the GENGT block all come from the MTM block for the location of Berlin, Germany, in this case.

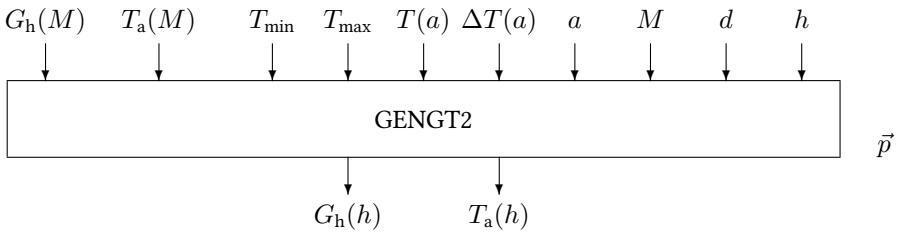
The PLOT block displays the generated time series of hourly ambient temperature data.



2.22 Block GENGT2

The GENGT2 block generates a series of hourly radiation and temperature data from monthly mean values. Optionally, hourly values of relative humidity may be generated. When a minute input is connected and different from zero, GENGT2 performs a linear interpolation between the two corresponding hours.

Internally, it simply calls GENGT which uses an old convention for timezone and longitude.



Name	GENGT2
Function	em0122
Inputs	10 . . . [12]
Outputs	3
Parameters	8 . . . [10]
Strings	0
Group	S

Inputs

1	Global horizontal irradiance $G_h(M)$ / W m^{-2} (monthly)
2	Ambient temperature $T_a(M)$ / °C (monthly)
3	Daily minimum ambient temperature T_{min} / °C
4	Daily maximum ambient temperature T_{max} / °C
5	Yearly average ambient temperature $T(a)$ / °C
6	Maximum ambient temperature difference $\Delta T(a)$ / °C
7	Year a
8	Month $M \in [1, 12]$
9	Day $d \in [1, 31]$
10	Hour $h \in [0, 23]$
11	Relative humidity $\varphi(M)$ (monthly)
12	Minute $m \in [0, 59]$

Outputs

1 Global radiation $G_{\mathrm{h}}(h)$ / W m^{-2}
2 Ambient temperature $T_{\mathrm{a}}(h)$ / °C
3 Relative humidity $\varphi_(h)$

Parameters

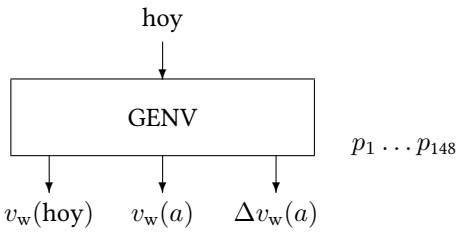
1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2 Longitude $\lambda \in [-180°, 180°]$, **east of Greenwich**; values west of Greenwich may be used with a minus sign
3 UTC Time zone $Z \in [-12, 12]$, Greenwich Mean Time $Z = 0$, Central European Time $Z = +1$.
4 Variance factor f_σ to the Gordon / Reddy correlation, eq **??**; if unknown $f_\sigma = 1$ is recommended
5 Coefficient c_σ corresponding to the year-to-year variability due to different climatic conditions. When c_σ is set to zero the year-to-year variability is omitted. $c_\sigma = 0.185$ approximates North American variability, while $c_\sigma = 0.3$ approximates European variability
6 Autocorrelation coefficient $\rho(1)$ at a lag of one day; if unknown $\rho(1) = 0.3$ is recommended
7 Autocorrelation coefficient $\rho(2)$ at a lag of two days; if unknown $\rho(2) = 0.57\rho(1)$ is recommended
8 Initialization I_{seed} of random number generator
9 Maximum allowed deviation between given monthly mean temperature and calculated value (0.1°C by default)
10 Maximum number of allowed iterations for temperature synthesis (2000 by default)

Strings

None

2.23 Block GENV

The GENV block generates hourly wind speed data for a specific site, based on the European Wind Atlas.



Name	GENV
Function	em0005
Inputs	1
Outputs	3
Parameters	148
Strings	0
Group	S

Inputs

1 Hour of year

Outputs

1 Hourly mean wind speed
2 Mean annual wind speed, calculated from BP(137..148)
3 Error of mean wind speed from generated values in per cent

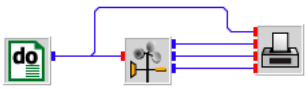
Parameters

1..96 Three-hour mean wind speed values
97..135 Power spectrum
136 Calibration factor
137..148 Monthly cubic mean wind speed data, uncalibrated

Strings

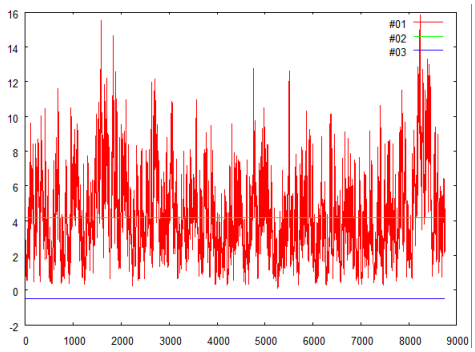
None

genv.vseit



A DO block runs through 8760 hours of a year and the GENV block calculates the wind speed time series for the location of Berlin, Germany.

The PLOT block displays the generated time series of hourly wind speed data, the annual mean wind speed and the deviation between the given annual mean wind speed and the corresponding generated mean value, which is below one percent, in this case.



Remark

The numerical method behind the GENV block is based on the European Wind Atlas. For about two hundred stations data have been published in the European Wind Atlas to satisfy the parameter requirements of the GENV block. These data can be found in the `resources/data/genv` directory of the INSEL installation directory.

It is not known, how well the method for the generation of wind speed data implemented in the GENV block can be used for Non-European sites.

Practical hints

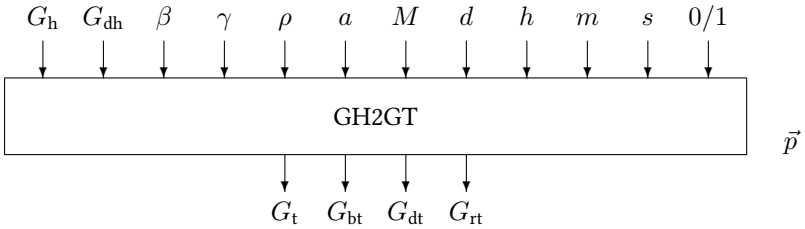
The example presented above is simple. The problem is, how the parameters of the GENV block can be determined.

Up to now March 2014) there is no convenient way to paste the data into a VSEit model. A workaround could be to use a text editor and copy one of the locations in the `genv` directory into a file similar to `genv.ins` which can be found in the `resources/data/genv` directory, save it and use the Import .ins file function of the VSEit interface. The GENV block can then be copied from the imported `.ins` file and pasted into your application.

2.24 Block GH2GT

Warning: This block is deprecated, because it used a west-positive convention for longitudes. Please use GH2GT2 instead, with east-positive longitudes.

The GH2GT block calculates the radiation on a tilted surface from horizontal data.



Name	GH2GT
Function	em0008
Inputs	9 . . . [12]
Outputs	4
Parameters	4
Strings	0
Group	S

Inputs

1	Global radiation horizontal G_h / W m^{-2}
2	Diffuse radiation horizontal G_{dh} / W m^{-2}
3	Slope β / °
4	Azimuth orientation of surface γ / °
5	Ground reflectance $\rho \in [0, 1]$
6	Year a (for example 2010)
7	Month $M \in [1, 12]$
8	Day $d \in [1, 31]$
9	Hour $h \in [0, 23]$
10	Minute $m \in [0, 59]$
11	Second $s \in [0, 60)$
12	Daylight saving time (0 no, 1 yes)

Outputs

1	Global radiation on tilted surface G_t / W m^{-2}
2	Beam radiation on tilted surface G_{bt} / W m^{-2}

3 Diffuse radiation on tilted surface G_{dt} / W m^{-2}
4 Reflected radiation on tilted surface G_{rt} / W m^{-2}

Parameters

1 Model
  0 Liu, Jordan
  1 Temps, Coulson
  2 Bugler, Hay, Kambezidis
  3 Klucher
  4 Hay
  5 Willmott
  6 Skartveit, Olseth
  7 Gueymard
  8 Perez et al.
  9 Reindl et al.
2 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
3 Longitude $\lambda \in [0°, 360°)$, Greenwich $= 0$, **west positive**
4 Time zone $Z \in [0, 23]$, e. g., Central European Time: $Z = 23$

Strings

None

Description

The first parameter determines how the diffuse radiation part is handled.

Liu, Jordan

In the Liu/Jordan model it is assumed that the diffuse radiation is isotropically distributed over the whole sky dome. The global radiation on a tilted surface is calculated by

$$G_t = G_{bh}\frac{\cos\theta}{\cos\theta_z} + \frac{1}{2}\,G_{dh}(1+\cos\beta) + \frac{1}{2}\,\rho\,G_h(1-\cos\beta)$$

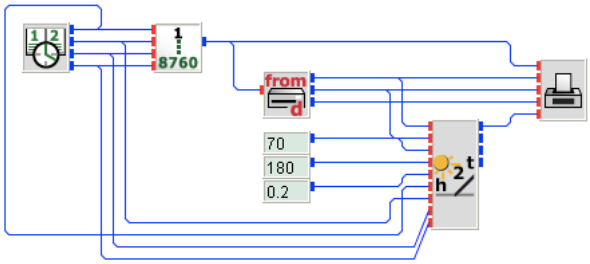
where

θ Incidence angle of beam radiation

θ_z Zenith angle

The first term of the sum defines G_{bt}, the second G_{dt} and the third one G_{rt}.

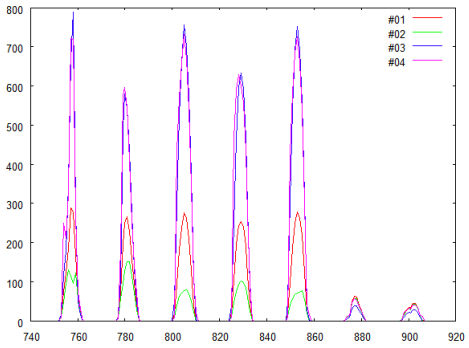
gh2gt.vseit



A CLOCK block runs through the first seven days of February 2010 in steps of one hour. The output signals are converted to the hour of the year which define the record number for the READD block and the x-coordinate of the PLOT block.

The READD blocks reads three values per record of the file `meteo82.dat` from the `examples/data` directory, namely the global radiation horizontal, the diffuse radiation and the tilted radiation facing south with a tilt of 70 degrees. These data are connected to the inputs of the GH2GT block and the PLOT block. The albedo is assumed constant with 20 percent.

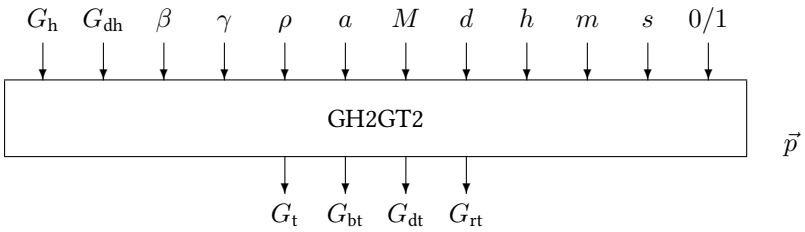
The PLOT block displays the data for comparison in the order global radiation horizontal, diffuse radiation, measured tilted radiation, and simulated tilted ratiation.



2.25 Block GH2GT2

The GH2GT2 block calculates the radiation on a tilted surface from horizontal data.

Internally, it simply calls GH2GT which uses an old convention for timezone and longitude.



Name	GH2GT2
Function	em0108
Inputs	9 . . . [12]
Outputs	4
Parameters	4
Strings	0
Group	S

Inputs

1	Global radiation horizontal G_h / W m^{-2}
2	Diffuse radiation horizontal G_{dh} / W m^{-2}
3	Slope β / °
4	Azimuth orientation of surface γ / °
5	Ground reflectance $\rho \in [0, 1]$
6	Year a (for example 2010)
7	Month $M \in [1, 12]$
8	Day $d \in [1, 31]$
9	Hour $h \in [0, 23]$
10	Minute $m \in [0, 59]$
11	Second $s \in [0, 60)$
12	Daylight saving time (0 no, 1 yes)

Outputs

1	Global radiation on tilted surface G_t / W m^{-2}
2	Beam radiation on tilted surface G_{bt} / W m^{-2}

3 Diffuse radiation on tilted surface G_{dt} / W m^{-2}
4 Reflected radiation on tilted surface G_{rt} / W m^{-2}

Parameters

1 Model
- 0 Liu, Jordan
- 1 Temps, Coulson
- 2 Bugler, Hay, Kambezidis
- 3 Klucher
- 4 Hay
- 5 Willmott
- 6 Skartveit, Olseth
- 7 Gueymard
- 8 Perez et al.
- 9 Reindl et al.

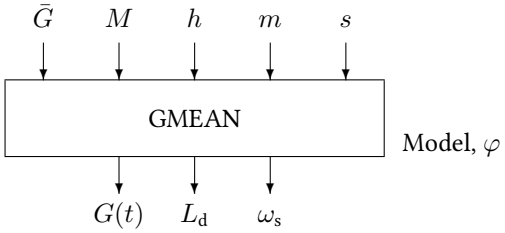
2 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
3 Longitude $\lambda \in [-180°, 180°]$, Greenwich $= 0$, **east positive**
4 UTC Time zone $Z \in [-12, 12]$, e. g., Central European Time: $Z = +1$

Strings

None

2.26 Block GMEAN

The GMEAN block returns the global radiation at a given real solar time for a mean monthly or daily radiation value based on statistical correlations.



Name	GMEAN
Function	em0013
Inputs	3 . . . [5]
Outputs	3
Parameters	2
Strings	0
Group	S

Inputs

1 Monthly or daily mean value $\bar{G}$ of global radiation / $\mathrm{W\,m^{-2}}$
2 Month M
3 Hour h
4 Minute m
5 Second s

Outputs

1 Mean global radiation $G(t)$ / $\mathrm{W\,m^{-2}}$
2 Mean daylength L_d / h
3 Mean sunrise hour angle ω_s / °

Parameters

1 Model
  0 Liu Jordan Collares-Pereira Rabl correlation
  1 World Bank Standard Solar Day
2 Latitude φ / °

Strings

None

Collares-Pereira Rabl correlation

The Collares-Pereira Rabl correlation which describes the ratio r between hourly and daily global irradiance for a given hour angle ω is given by

$$r = \frac{\pi}{24} \frac{\cos\omega - \cos\omega_{ss}}{\sin\omega_{sr} - \omega_{ss}\cos\omega_{ss}}(a + b\cos\omega)$$

with the coefficients

$$a = 0.4090 + 0.5016\sin(\omega_{ss} - 60)$$
$$b = 0.6609 - 0.4767\sin(\omega_{ss} - 60)$$

The relation between sunset hour angle ω_{ss} and day length (in hours) is $\omega_{ss} = 7.5(L_d - 12) + 90$.

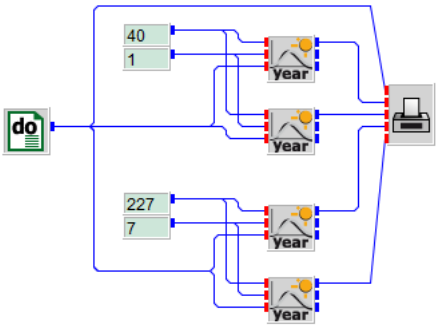
Standard solar day

The standard solar day has been introduced by the World Bank and is defined through the equation

$$G(t) = G_{max}\sin\left(\frac{\pi t}{L_d}\right) \qquad t = 0, 1, \ldots 11\ \text{h}$$

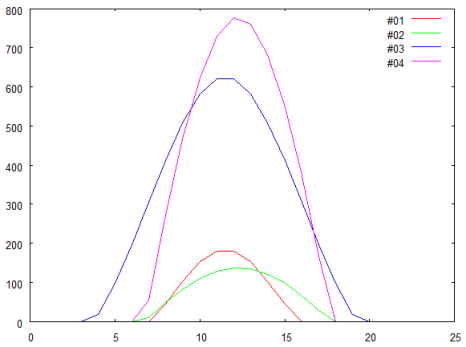
G_{max} denotes the daily maximum of the global radiation on a horizontal surface. L_d stands for the day length and is assumed constant with 11 hours. Both, hour angle and sunset hour angle are measured from true solar noon in this case.

gmean.vseit



A DO block counts the daily hours from zero to 23 in steps of one. Four CONST blocks provide the months January and July and their mean irradiances of 40 and 227 W/m^2, respectively. Four GMEAN blocks calculate the hourly mean irradiances from the Collarea-Pereira Rabl (upper GMEAN block) and Standard Solar Days (lower GMEAN block) correlations.

A PLOT block displays the four resulting curves on screen.



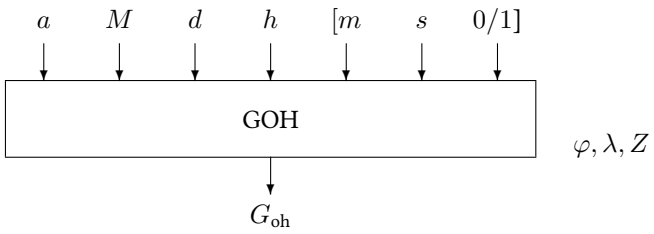
Remark Please observe, that the Standard Solar Day assumes a constant day length of eleven hours, which is automatically centered around true solar noon by the GMEAN block. The Collares-Pereira Rabl block adds thirty minutes to the hour input as suggested by the Collares-Pereira Rabl approach.

2.27 Block GOH

Warning: The explicit usage of this block is deprecated, because it uses a west-positive convention for longitudes.

Please use GOH2 instead, with east-positive longitudes. GOH is still used internally by GOH2.

The GOH block returns the radiation outside atmosphere on a horizontal surface.



Name	GOH
Function	em0012
Inputs	4 . . . [7]
Outputs	1
Parameters	3
Strings	0
Group	S

Inputs

1	Year a (for example 2001)
2	Month $M \in [1, 12]$
3	Day $d \in [1, 31]$
4	Hour $h \in [0, 23]$
5	Minute $m \in [0, 59]$
6	Second $s \in [0, 60)$
7	Daylight saving time (0 no, 1 yes)

Outputs

1	Radiation outside the atmosphere on a horizontal surface G_{oh} / $\mathrm{W\,m^{-2}}$

Parameters

1	Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2	Longitude $\lambda \in [0°, 360°)$, Greenwich $= 0°$, west positive

3	Time zone $Z \in [0, 23]$, e. g., Central European Time: $Z = 23$	
Strings		
	None	

Description The radiation outside the atmosphere on a horizontal surface G_{oh} is given by

$$G_{oh} = \epsilon_0 G_{sc} \cos\theta_z$$

where

ϵ_0	Excentricity factor $(r_0/r)^2$, with mean distance between Sun and Earth r_0 $(1.496 \times 10^{11}\,\mathrm{m})$ and time dependent distance between Sun and Earth r
G_{sc}	Solar constant $(1367\,\mathrm{W/m^2})$
θ_z	Zenith angle

The excentricity factor is calculated by Spencer's Fourier series

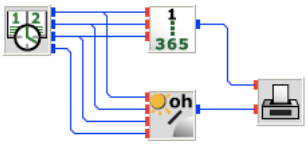
$$\begin{aligned}\epsilon_0 \;=\; & 1.00010 + 0.034221\cos\Gamma + 0.001280\sin\Gamma \\ & + 0.000719\cos 2\Gamma + 0.000077\sin 2\Gamma\end{aligned}$$

where

Γ	day angle $(2\pi\,(\mathrm{doy}-1)/365)$, with the day of year $\mathrm{doy} \in [1, 365]$; leap years are not considered

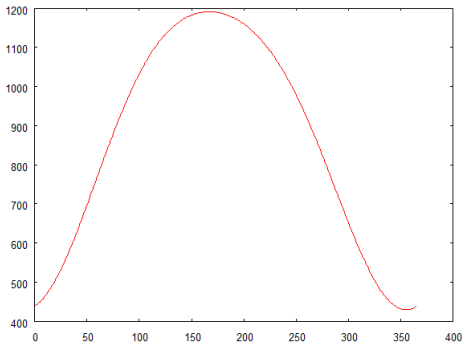
The zenith angle θ_z is calculated by an internal call to block SUNAE (using the Hollands/Maier model) and block INANG ($\beta = 0°, \gamma = 180°$).

goh.vseit



A CLOCK block delivers year, month, day, and hour data for one year in steps of one day. The hour is fixed to 12. The block GOH calculates the correspondig value of the extraterrestrial radiation for the location of Stuttgart, Germany. The DOY block converts the outputs of the CLOCK block to the day of the year.

The PLOT block plots the time series.

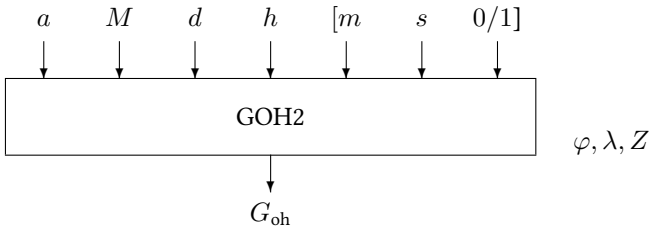


See also Block GON, block SUNAE.

2.28 Block GOH2

The GOH2 block returns the radiation outside atmosphere on a horizontal surface.

Internally, it simply calls GOH which uses an old convention for timezone and longitude.



Name	GOH2
Function	em0112
Inputs	4 . . . [7]
Outputs	1
Parameters	3
Strings	0
Group	S

Inputs

1	Year a (for example 2021)
2	Month $M \in [1, 12]$
3	Day $d \in [1, 31]$
4	Hour $h \in [0, 23]$
5	Minute $m \in [0, 59]$
6	Second $s \in [0, 60)$
7	Daylight saving time (0 no, 1 yes)

Outputs

1	Radiation outside the atmosphere on a horizontal surface G_{oh} / W m^{-2}

Parameters

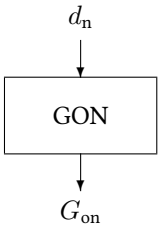
1	Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2	Longitude $\lambda \in [-180°, 180°]$, Greenwich $= 0°$, west positive
3	UTC Time zone $Z \in [-12, 12]$, e. g., Central European Time: $Z = +1$

Strings

None

2.29 Block GON

The GON block calculates the normal extraterrestrial radiation.



model

Name	GON
Function	em0002
Inputs	1
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Day of the year d_n

Outputs

1 Direct normal radiation outside atmosphere G_{on} / W m^{-2}

Parameters

1 Approximation
- 0 Spencer
- 1 Duffie Beckman

Strings

None

Description

The radiation outside the atmosphere normal to the Sun G_{on} is given by

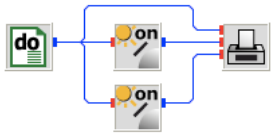
$$G_{on} = \epsilon_0 G_{sc}$$

where

ϵ_0	Excentricity factor $(r_0/r)^2$, with mean distance between Sun and Earth r_0 $(1.496 \times 10^{11}\ \mathrm{m})$ and time dependent distance between Sun and Earth r
G_{sc}	Solar constant $(1367\ \mathrm{W/m^2})$

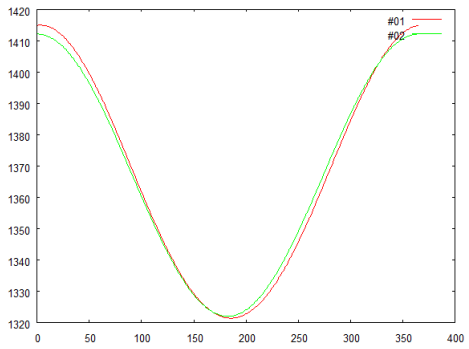
For the calculation of the excentricity factor ϵ_0 please refer to block GOH.

gon.vseit



A DO block counts through the days of a year from one to 365. The two GON calculate the extraterrestrial radiation normal to the Sun for the location of Stuttgart, Germany. The upper block uses the Spencer approximation, the lower one the excentricty as given by Duffie and Beckman.

The PLOT block plots the time series.

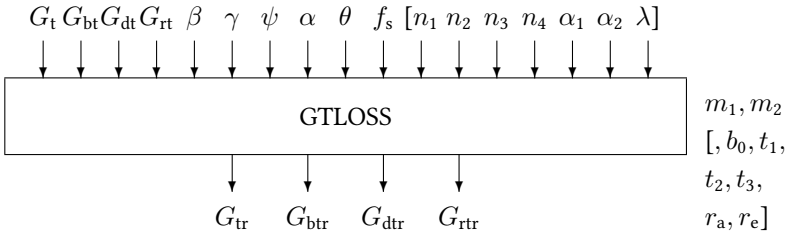


See also

Block GOH, block SUNAE.

2.30 Block GTLOSS

The GTLOSS block calculates losses of radiation incident on a tilted solar module. Four different models can be used to calculate the reflection and eventually absorption losses due to incidence angles of the light not equal to zero. The block provides two models to calculate the losses of the diffuse radiation.



Name	GTLOSS
Function	em0004
Inputs	10 . . . [17]
Outputs	5
Parameters	2 . . . [8]
Strings	0
Group	S

Inputs

1	Global radiation on tilted surface G_t / W m^{-2}
2	Beam radiation on tilted surface G_{bt} / W m^{-2}
3	Diffuse radiation from sky on tilted surface G_{st} / W m^{-2}
4	Diffuse radiation from ground on tilted surface G_{rt} / W m^{-2}
5	Slope β / °
6	Azimuth of module γ / °
7	Azimuth of the Sun ψ (north $= 0°$, east $= 90° \ldots$ west $= 270°$)
8	Elevation of the Sun α / °
9	Incidence angle θ / °
10	Soil loss factor f at $\theta = 0$
11	Refraction index of layer 1 (usually glass) n_1
12	Refraction index of layer 2 (e. g., EVA) n_2
13	Refraction index of layer 3 (e. g., TiO_2) n_3
14	Refraction index of layer 4 (usually Si) n_4
15	Absorption coefficient of layer 1 α_1 / m^{-1}

16 Absorption coefficient of layer 2 α_2 / m^{-1}
17 Wavelength λ / nm

Outputs

1 Global radiation on tilted surface with reflection G_{tr} / $\mathrm{W\,m}^{-2}$
2 Beam radiation on tilted surface with reflection G_{btr} / $\mathrm{W\,m}^{-2}$
3 Diffuse radiation from sky on tilted surface with reflection G_{str} / $\mathrm{W\,m}^{-2}$
4 Diffuse radiation from ground on tilted surface with reflection G_{rtr} / $\mathrm{W\,m}^{-2}$
5 Effective transmissivity

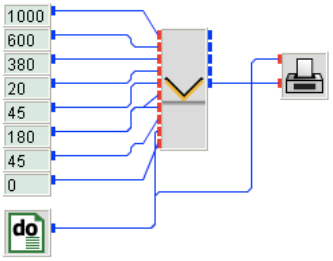
Parameters

1 Model of beam radiation m_1
  0 Sjerps-Koomen / Alsema
  1 ASHRAE
  2 Krauter
  3 Lanzerstorfer
2 Model of diffuse radiation from sky m_2
  0 Duffie / Beckmann
  1 DIN 5034
3 ASHRAE Parameter (0.05 is a good guess)
4 Thickness of layer 1 t_1 / m
5 Thickness of layer 2 t_2 / m
6 Thickness of layer 3 t_3 / nm
7 Number of segments for integration over azimuth r_{a}
8 Number of segments for integration over elevation r_{e}

Strings

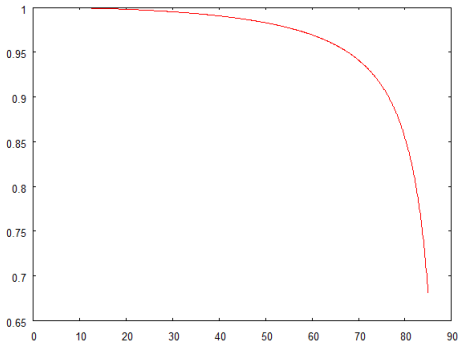
None

gtloss.vseit



The example shows an application of the most simple loss mechanism simulation based on the ASHRAE model. All inputs are set constant except the incidence angle which is varied between zero and 85 degrees by a DO block.

A PLOT block shows the effective transmittance as a function of the incidence angle.

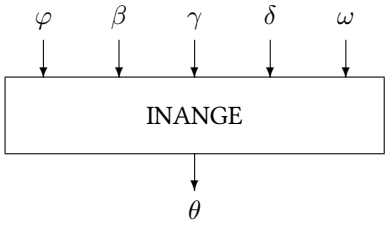


For further examples, please refer to the `examples` directory.

Remark The GTLOSS block is still in an experimental stage and will probably be redesigned in the future.

2.31 Block INANGE

The INANGE block calculates the incidence angle of beam radiation on an arbitrarily oriented plane, when the position of the Sun is given in equator coordinates.



Name	INANGE
Function	em0007
Inputs	5
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2 Slope β
3 Azimuth angle of the surface γ (north $= 0°$, east $= 90° \ldots$ west $= 270°$)
4 Declination of the Sun δ / °
5 Hour angle of the Sun ω ($0° \ldots 360°$west)

Outputs

1 Incidence angle θ / ° between beam radiation and surface normal

Parameters

None

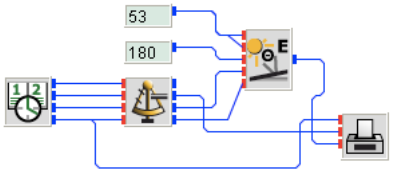
Strings

None

Description The incidence angle θ is calculated from the equation

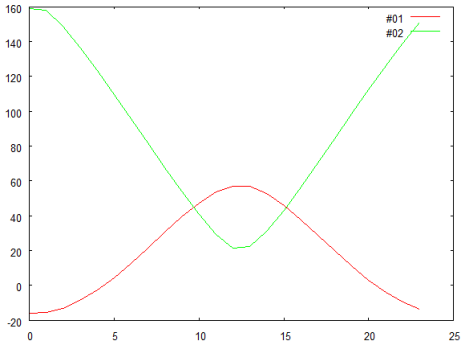
$$\begin{aligned}\cos\theta \;=\; & \sin\delta\,(\sin\varphi\;\cos\beta+\cos\varphi\;\sin\beta\;\cos\gamma)\\ & +\cos\delta\;\cos\omega\,(\cos\varphi\;\cos\beta-\sin\varphi\;\sin\beta\;\cos\gamma)\\ & -\cos\delta\;\sin\beta\;\sin\gamma\;\sin\omega\end{aligned}$$

inange.vseit



A CLOCK block runs through the 23rd of May 2010 in steps of one hour. A SUNAE block calculates the position of the Sun for a latitude of 53 degrees, 8 degrees east (that is close to Oldenburg, Germany).

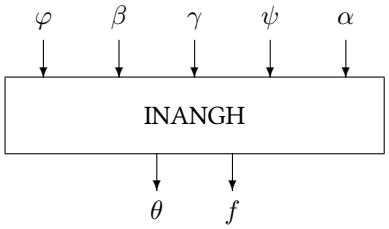
Assuming latitude and tilt angle equal to 53 degrees with a surface facing south (azimuth 180 degrees) the INANGE block calculates the incidence angle using equator coordinates and the PLOT block displays the elevation of the Sun and the incidence angle as a function of the hour of the day.



See also Block INANGH.

2.32 Block INANGH

The INANGH block calculates the incidence angle of beam radiation on an arbitrarily oriented plane, when the position of the Sun is given in horizon coordinates.



Name	INANGH
Function	em0007
Inputs	5
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2 Slope β / °
3 Azimuth angle of the surface γ (north $= 0°$, east $= 90° \ldots$ west $= 270°$)
4 Azimuth angle of the Sun ψ / °
5 Elevation of the Sun α / °

Outputs

1 Incidence angle θ / ° between beam radiation and surface normal
2 Flag f set to one, when Sun is over the horizon, set to zero otherwise

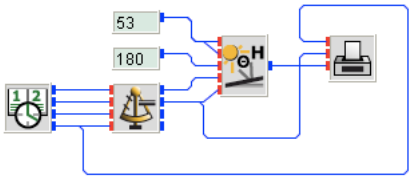
Parameters

None

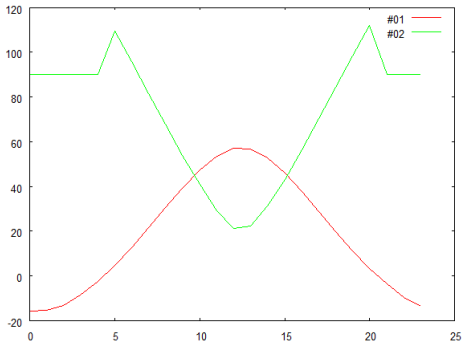
Strings

None

inangh.vseit



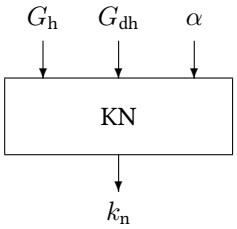
This is exactly the same example as the one presented for INANGE with the exception that now the calculation of the Solar position uses horizon coordinates. Surprising enough, the graph displayed by the PLOT block is different for the night hours.



See also Block INANGE.

2.33 Block KN

The KN block calculates the nebulosity index.



Name	KN
Function	em0030
Inputs	3
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Global radiation horizontal / $\mathrm{W\,m^{-2}}$
2 Diffuse radiation horizontal / $\mathrm{W\,m^{-2}}$
3 Elevation / °

Outputs

1 Nebulosity index k_n

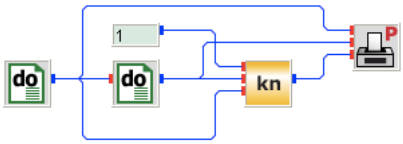
Parameters

None

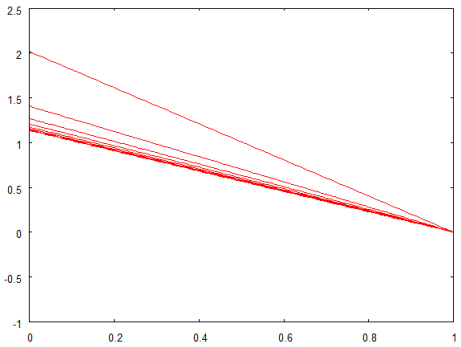
Strings

None

kn.vseit

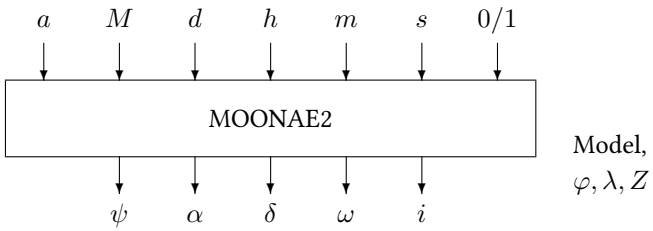


Two nested DO blocks vary the elevation of the Sun between zero and 90 degrees in steps of ten (outer DO block) and a normalised diffuse ratiation between zero and one. The global irradiance is normalised to one, too. The KN calculates the nebulosity indices, which are displayed by a PLOTP block.



2.34 Block MOONAE2

The MOONAE2 block calculates the position of the Moon in horizon and equator coordinates.



Name	MOONAE2
Function	em0040
Inputs	6
Outputs	5
Parameters	3
Strings	0
Group	S

Inputs

1 Year a (for example 2021)
2 Month $M \in [1, 12]$
3 Day $d \in [1, 31]$
4 Hour $h \in [0, 23]$
5 Minute $m \in [0, 59]$
6 Second $s \in [0, 60)$

Outputs

1 Azimuth ψ (north $= 0°$, east $= 90° \ldots$ west $= 270°$)
2 Elevation α / °
3 Declination δ / °
4 Hour angle ω ($0° \ldots 360°$west)
5 Moon phase angle i ($0°$: full moon, $90°$: half-moon, $180°$: new moon)

Parameters

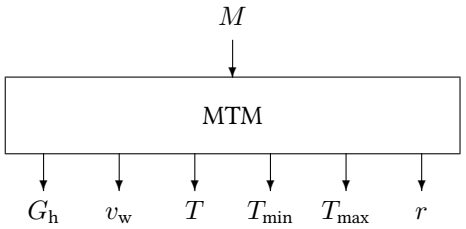
1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2 Longitude $\lambda \in [0°, 360°)$, Greenwich $= 0°$, **east positive**
3 UTC-Time zone $Z \in [-12, 12]$, e. g. Central European Time: $Z = +1$

Strings

None

2.35 Block MTM

The MTM block returns monthly mean values of meteorological data from the inselWeather database.



$\varphi, \lambda, \Delta\varphi, \Delta\lambda,$
CC, KC, location

Name	MTM
Function	em0018
Inputs	1
Outputs	9
Parameters	0 . . . [6]
Strings	1
Group	S

Inputs

1 Month $M \in [1, 12]$

Outputs

1 Global radiation G_h / W m^{-2} on a horizontal plane
2 Wind speed v_w / m s^{-1}
3 Ambient temperature T / °C
4 Minimum ambient temperature $T_{a,min}$ / °C
5 Maximum ambient temperature $T_{a,max}$ / °C / °C
6 Rain / mm
7 Annual mean ambient temperature / °C
8 Maximum ambient temperature difference / °C
9 Relative humidity

Parameters

1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
2 Longitude $\lambda \in [0°, 360°)$, west of Greenwich; values east of Greenwich may be used with a minus sign
3 Latitude range $\Delta\varphi$ / °

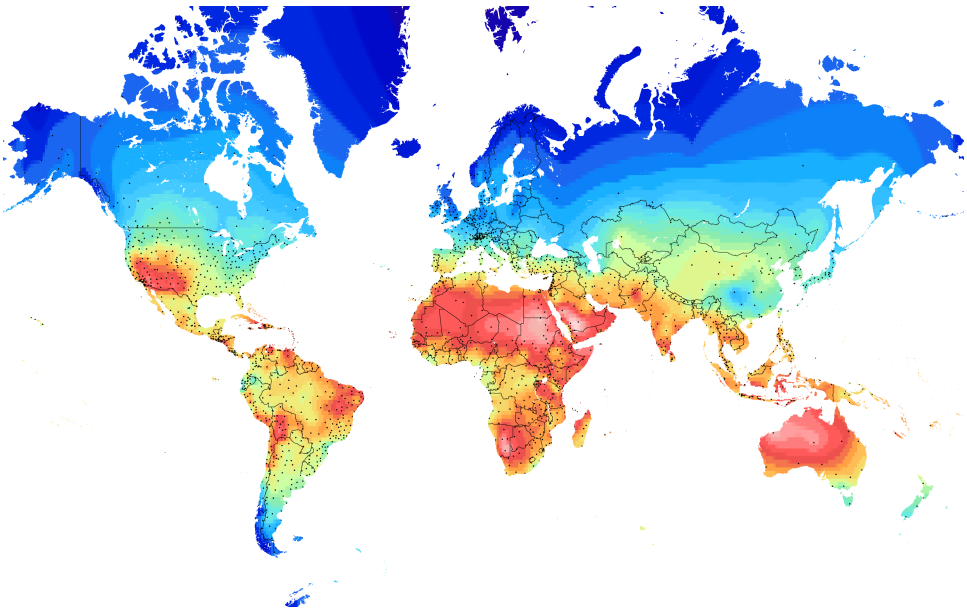
4 Longitude range $\Delta\lambda$ / °
5 Country code CC
6 Continent code KC

Strings

1 Name of location

Description

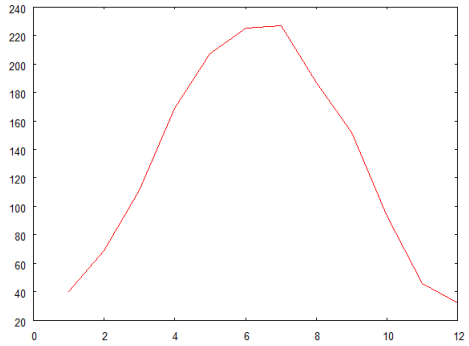
The MTM database provides long-term monthly mean values of the most important climatological parameters like global irradiance, ambient temperature etc. The image below shows the interpolated mean annual irradiance data distribution on a horizontal surface as stored for about 2 000 locations worldwide.



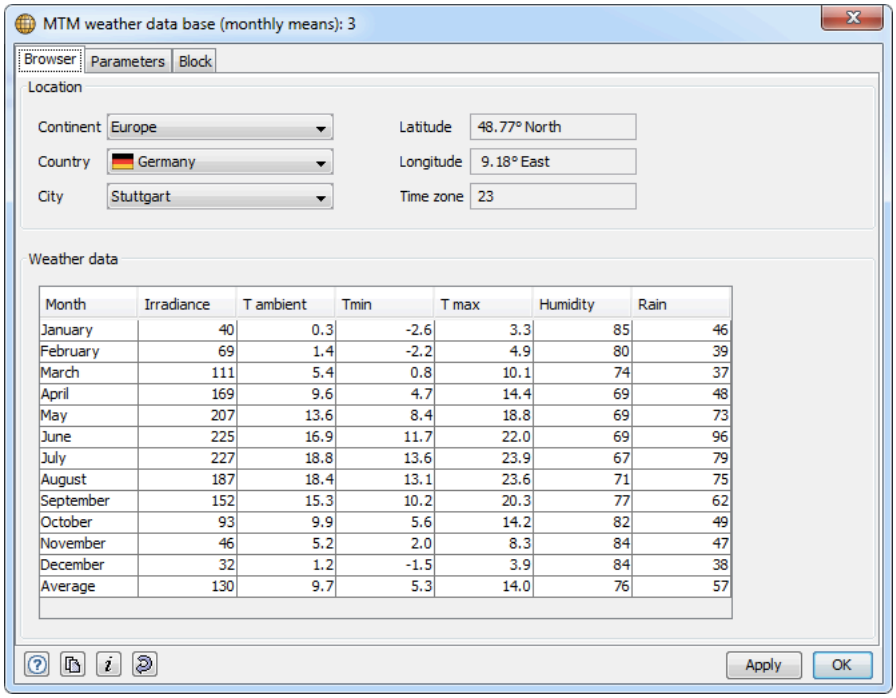
mtm.vseit

A DO block counts through the twelve months of an arbitrary year. The month is connected to a MTM block which delivers irradiance data for the location of Stuttgart, Germany.

A PLOT block displays the data.



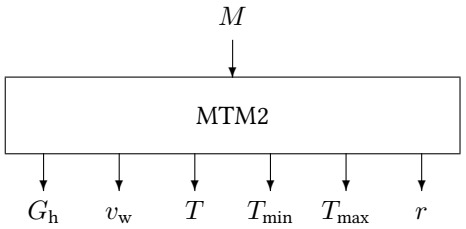
The MTM block's VSEit entity editor provides a browser to access the data base.



Month	Irradiance	T ambient	Tmin	T max	Humidity	Rain
January	40	0.3	-2.6	3.3	85	46
February	69	1.4	-2.2	4.9	80	39
March	111	5.4	0.8	10.1	74	37
April	169	9.6	4.7	14.4	69	48
May	207	13.6	8.4	18.8	69	73
June	225	16.9	11.7	22.0	69	96
July	227	18.8	13.6	23.9	67	79
August	187	18.4	13.1	23.6	71	75
September	152	15.3	10.2	20.3	77	62
October	93	9.9	5.6	14.2	82	49
November	46	5.2	2.0	8.3	84	47
December	32	1.2	-1.5	3.9	84	38
Average	130	9.7	5.3	14.0	76	57

2.36 Block MTM2

The MTM2 block returns monthly mean values of meteorological data from the inselWeather database.



$\varphi, \lambda, \Delta\varphi, \Delta\lambda,$
CC, KC, location

Name	MTM2
Function	em0118
Inputs	1
Outputs	9
Parameters	0 . . . [6]
Strings	1
Group	S

Inputs

1	Month $M \in [1, 12]$

Outputs

1	Global radiation G_h / W m^{-2} on a horizontal plane
2	Wind speed v_w / m s^{-1}
3	Ambient temperature T / °C
4	Minimum ambient temperature $T_{a,min}$ / °C
5	Maximum ambient temperature $T_{a,max}$ / °C / °C
6	Rain / mm
7	Annual mean ambient temperature / °C
8	Maximum ambient temperature difference / °C
9	Relative humidity

Parameters

None

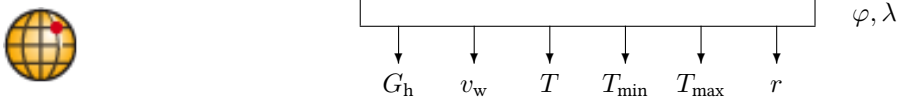
Strings

1	Name of location

2.37 Block MTMLALO

Warning: The explicit usage of this block is deprecated, because it uses a west-positive convention for longitudes. Please use MTMLALO2 instead, with east-positive longitudes. MTMLALO is still used internally by MTMLALO2.

The MTMLALO block returns monthly mean values of meteorological data for a location specified by latitude and longitude interpolated from the inselWeather database.



Name	MTMLALO
Function	em0018
Inputs	1
Outputs	9
Parameters	2
Strings	0
Group	S

Inputs

1 Month $M \in [1, 12]$

Outputs

1 Global radiation G_h / W m^{-2} on a horizontal plane
2 Wind speed v_w / m s^{-1}
3 Ambient temperature T / °C
4 Minimum ambient temperature $T_{a,min}$ / °C
5 Maximum ambient temperature $T_{a,max}$ / °C / °C
6 Rain / mm
7 Annual mean ambient temperature / °C
8 Maximum ambient temperature difference / °C
9 Relative humidity

Parameters

1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive

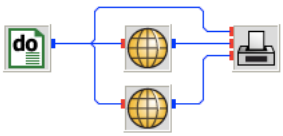
2 Longitude $\lambda \in [0°, 360°)$, west of Greenwich; values east of Greenwich may be used with a minus sign

Strings

None

---

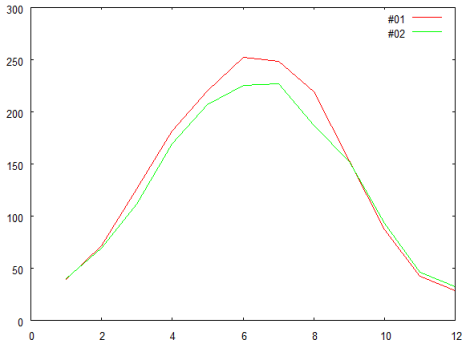
mtmlalo.vseit



The database behind the MTM block can also be accessed via the MTMLALO block. In this case, values for latitude and longitude need to be provided as parameters. The MTMLALO block looks for the three closest locations and interpolates their data weighted by the distance. If the neighbors are very far away, the block displays a warning message.

The example uses the latitude 48 degrees north and nine degrees east (i. e., -9) which is close to Stuttgart, Germany.

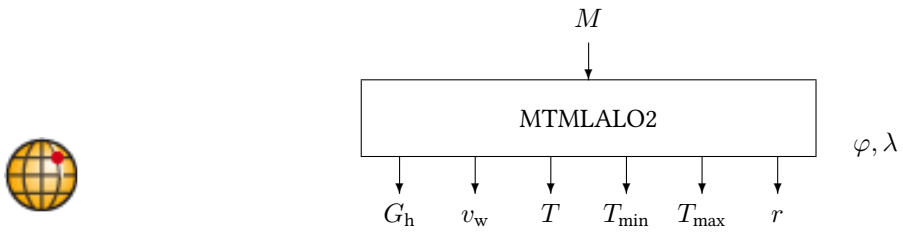
The Stuttgart data from the MTM block (lower block) are displayed for comparison by the PLOT block.



2.38 Block MTMLALO2

The MTMLALO2 block returns monthly mean values of meteorological data for a location specified by latitude and longitude interpolated from the inselWeather database.

Internally, it simply calls MTMLALO which uses an old convention for timezone and longitude.



Name	MTMLALO2
Function	em0118
Inputs	1
Outputs	9
Parameters	2
Strings	0
Group	S

Inputs

1 Month $M \in [1, 12]$

Outputs

1 Global radiation G_h / W m^{-2} on a horizontal plane
2 Wind speed v_w / m s^{-1}
3 Ambient temperature T / °C
4 Minimum ambient temperature $T_{a,min}$ / °C
5 Maximum ambient temperature $T_{a,max}$ / °C / °C
6 Rain / mm
7 Annual mean ambient temperature / °C
8 Maximum ambient temperature difference / °C
9 Relative humidity

Parameters

1 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive

2	Longitude $\lambda \in [-180°, 180°]$, east of Greenwich; values west of Greenwich may be used with a minus sign

Strings

None

2.39 Block MTMUP

The MTMUP block udates the INSEL weather database.

Update file name

Name	MTMUP
Function	em0026
Inputs	0
Outputs	0
Parameters	0
Strings	1
Group	C

Inputs

None

Outputs

None

Parameters

None

Strings

1 Name of the update file

Description

The following text is a valid control file and should explain the method used by the MTMUP block. A user interface is not yet available in INSEL.

```
# Control file for updating the MTM data base
# To add sites to the data base, the following data must be given
#
#   Site      = Nname of site
#   Continent = Code number of continent
#               1 = Europe
#               2 = Asia
```

```
#               3 = Africa
#               4 = USA
#               5 = Latin America (see file MT.KEY)
#   Country   = Code number of country; see data base file MT.CC;
#               87 = Germany
#   Latitude  = Latitude of the site
#   Longitude = Longitude of the site
#   Command   = NEW, MODIFY or DELETE
#
# and at least one of the following data records should be given
# for each site
#
#   Radiation   = 12 monthly mean values of radiation
#   Wind        = 12 monthly mean values of wind speed
#   Temperature = 12 monthly mean values of temperature
#   MinTemperat = 12 monthly minimum temperature values
#   MaxTemperat = 12 monthly maximum temperature values
#   Rain        = 12 monthly mean values of rain fall
#   Humidity    = 12 monthly mean values of relative humidity
#
# Empty lines are ignored.
# Text following the characters "#", "%" and ";" are ignored to give the
# possibility of commenting.
# The first option of every data set must be "Site= ...".
# All other options ("Continent= ...", "Country=...", "Latitude=...", ...)
# up to the next option "Site= ..." or the end of file belong to the site.
# A end of file may be simulated with two successive lines containing the
# text "RETURN"
# The meaning of COMMAND is:
#
#   COMMAND = NEW:    Site may exist, data may not exist and data record
#                     will be created. If site not exists, site record
#                     will be created. If data already exists, an error
#                     message will be displayed and excecution stops.
#   COMMAND = MODIFY: Site may exist, data may exist.
#                     If site and/or data exist, they will be overwritten
#                     otherwise new records will be created
#   COMMAND = DELETE: Delete given data records of one site. If no more
#                     data records are available for the given site, the
#                     site will be deleted.
#   COMMAND = LIST:   List data of given site. If site does not exist,
#                     an error message will be displayed and excecution
#                     stops.
#
#
# Example: 1
#
#   Site      = Aurich
#   Continent =   1    # Europe
#   Country   =  87    # Germany
#   Latitude  =  53.2  # test
#   Longitude = 350.8
#   Command   = Modify
#   # Month       1    2    3    4    5    6    7    8    9   10   11   12
```

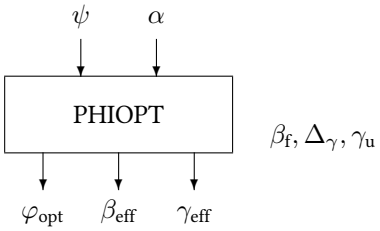
```
#   Radiation   = 11   12   13   14   15   16   16   15   14   13   12   11
#   Wind        = 13   15   15   18   30   40   17   10   25   24   43   29
#   Temperature = -3.0 -5.0  8.0 12.0 15.0 20.0 25.9 29.0 20.0 15.0 10.0  4.0
#   MinTemperat = -7.0 -9.0  0.0  7.0  9.0 15.0 14.9 20.0 13.0 10.0  7.0  0.0
#   MaxTemperat =  2.0  2.0 12.0 16.0 20.0 22.0 28.9 33.0 25.0 20.0 17.0 11.0
#   Rain        = 113  115  115  118  130  140  117  110  125  124  143  129
#   RETURN
#   RETURN
#
#
# Example: 2
#
#   Site      = Aurich
#   Continent =   1    # Europe
#   Country   =  87    # Germany
#   Latitude  =  53.2  # test
#   Longitude = 350.8
#   Command   = delete
#   # Month       1    2    3    4    5    6    7    8    9   10   11   12
#   Radiation =
#   Wind      =
#   RETURN  # Remark: In this case, if there are still some data records
#   RETURN  # given in the data base for the site, the site will not be
#           # deleted. If the last data record for the site gets deleted,
#           # the site will be erased from the data base

Site      = Stuttgart
Command   = Modify
Humidity  = 0 0 0 0 0 0 0 0 0 0 0 0

Site      = Torino
Command   = Modify
Humidity  = 0 0 0 0 0 0 0 0 0 0 0 0

Site      = Aurich
Command   = Modify
Humidity  = 0 0 0 0 0 0 0 0 0 0 0 0

Site      = Bremen
Command   = Modify
Humidity  = 0 0 0 0 0 0 0 0 0 0 0 0
```

mtmup.vseit

The MTMUP block is the only block in the model to be executed.

2.40 Block PHIOPT

The PHIOPT block calculates the optimum rotation angle for a one-axis tracked receiver.



Name	PHIOPT
Function	em0038
Inputs	2
Outputs	3
Parameters	3
Strings	0
Group	S

Inputs

1 Solar azimuth ψ / °
2 Solar elevation α / °

Outputs

1 Optimum rotation angle φ_{opt} / °
2 Effective tilt angle β_{eff} / °
3 Effective azimuth angle γ_{eff} / °

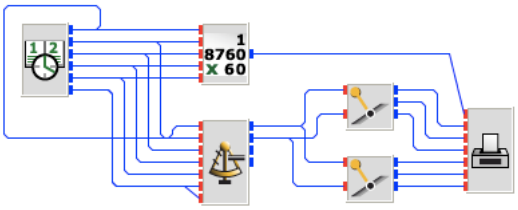
Parameters

1 Tilt angle of fixed rotation axis β_f / °
2 Deviation of axis from north/south direction (clockwise positive) Δ_γ / ° $\in [-90, 90]$
3 Azimuth angle if undefined γ_u / °
4 Mechanical limit towards east (degrees < 0, zero defaults to -90)
5 Mechanical limit towards west (degrees > 0 zero defaults to $+90$)

Strings

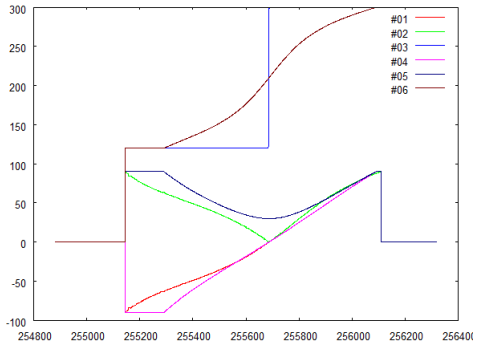
None

phiopt.vseit In this example two tracking systems with different orientations are compared.



The first (upper PHIOPT block) simulates a horizontal rotation axis, the other (lower PHIOPT block) an axis with a tilt angle of 30 degrees. In both cases it is assumed that the deviation from north/south direction is +30 degrees, i. e., approximately south/west.

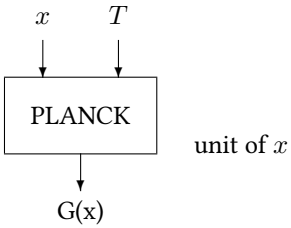
The model is controlled by a CLOCK block which runs through the 27th of June 2010 in steps of one minute. The SUNAE block is used to calculate the position of the Sun as required by the PHIOPT blocks. The MOY block is used as convenient x-coordinate for the PLOT block.



The six curves show the triples optimum rotation angle (01 and 04), effective tilt angle (02 and 05) and effective azimuth angle (03 and 06). As could be expected both tracking systems face the eastern direction in the morning hours and west in the afternoon (red and magenta curves). The effective tilt angles start with 90 degrees with a tendency towards low values around noon. Due to the asymmetry of the rotation axis with respect to south the azimuth angles vary between 120 and 300 degrees, respectively.

2.41 Block PLANCK

The PLANCK block simulates Planck's radiation law.



unit of x

Name	PLANCK
Function	em0001
Inputs	2
Outputs	3
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Wavelength λ/μm or photon energy / ev, depending on BP(1)
2 Blackbody temperature T/K

Outputs

1 Blackbody radiation after Planck If $s = 0$ $G(x)/\mathrm{W\,m^{-2}\,\mu m^{-1}}$, if $s = 1$ $G(x)/\mathrm{W\,m^{-2}\,eV^{-1}}$
2 Blackbody radiation after Wien
3 Blackbody radiation after Rayleigh-Jeans

Parameters

1 Unit switch s. If $s = 0$ (default) input one is interpreted as wavelength in μm, if $s = 1$ input one is interpreted as energy in eV.

Strings

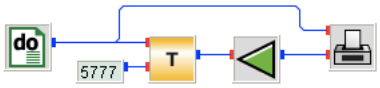
None

Description The radiation law known as Planck's law is given by

$$G(\lambda) = 2\pi hc^2 \frac{1}{(\lambda^5 \exp(\frac{hc}{\lambda kT}) - 1)}$$

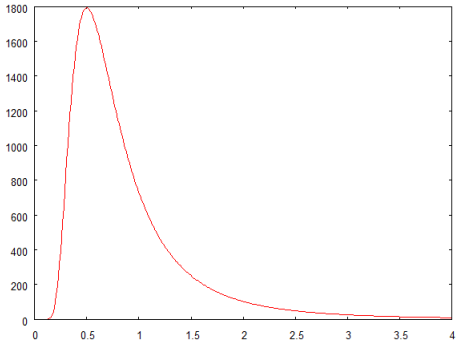
where λ denotes the wavelength, h is the Planck constant $6.6260755 \cdot 10^{-34}$ J s, c is the speed of light in vacuum 299 792 458 m s^{-1}, k is the Boltzmann constant $1.380658 \cdot 10^{-23}$ J K^{-1} and T the temperature of the body in kelvin.

planck.vseit



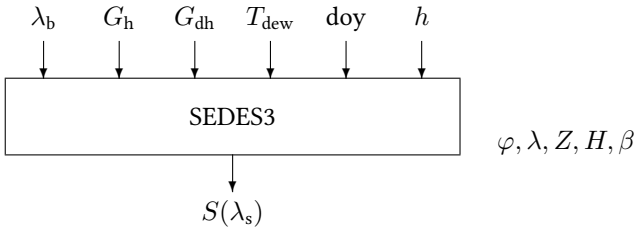
A DO block generates wavelength values from 0 to 4 μm in steps of 0.01 μm. A CONST block provides the black body temperature of the Sun. The PLANCK block calculates the spectral distribution, which is attenuated by the dilution factor 46200.

The PLOT block displays the solar spectrum outside atmosphere.



2.42 Block SEDES3

The SEDES3 block calculates solar irradiance spectra for the range between 0.3 and 1.4 micrometer.



Name	SEDES3
Function	em0033
Inputs	6
Outputs	1
Parameters	5
Strings	0
Group	S

Inputs

1 Wavelength λ_s/ μm
2 Global radiation horizontal G_h / W m^{-2}
3 Diffuse radiation horizontal G_{dh} / W m^{-2}
4 Dewpoint temperature T_{dew} / °C
5 Day of the year doy
6 Hour h

Outputs

1 Spectral density $S(\lambda_s)$/ W m^{-2} μm^{-1}

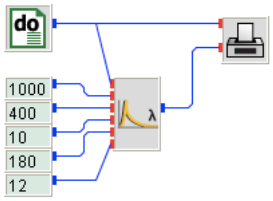
Parameters

1 Latitude φ / ° (north positive)
2 Longitiude λ / ° (east positive)
3 UTC Time zone Z
4 Height above sea level H / m
5 Tilt angle β / °

Strings

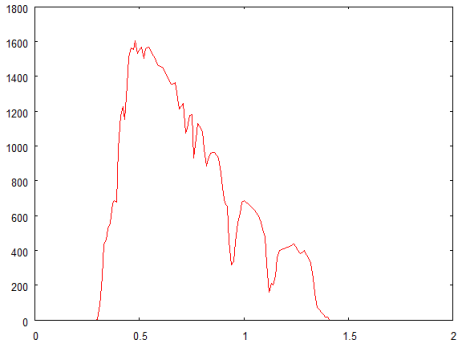
None

sedes3.vseit



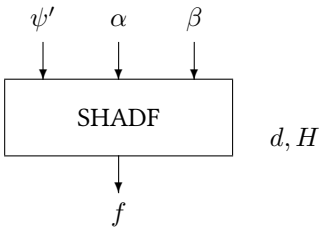
A DO block varies wavelengths between zero and 2 micrometers in steps of 0.01. All other inputs to the SEDES3 block are set constant.

The PLOT block displays the spectrum on screen.



2.43 Block SHADF

The SHADF block calculates the fraction of unshaded area of rows of tilted receivers.



Name	SHADF
Function	em0031
Inputs	3
Outputs	1
Parameters	2 . . . [3]
Strings	0
Group	S

Inputs

1 Difference between surface azimuth and azimuth of the Sun ψ' / °
2 Solar elevation α / °
3 Tilt angle β / °

Outputs

1 Fraction of unshaded area f

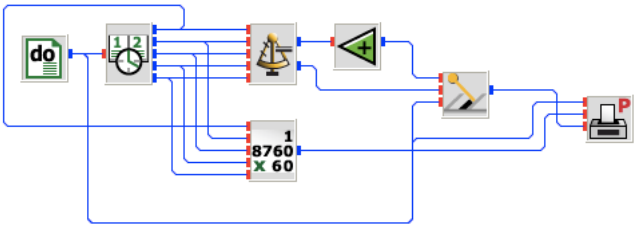
Parameters

1 Distance between module rows d / m
2 Height of modules h / m

Strings

None

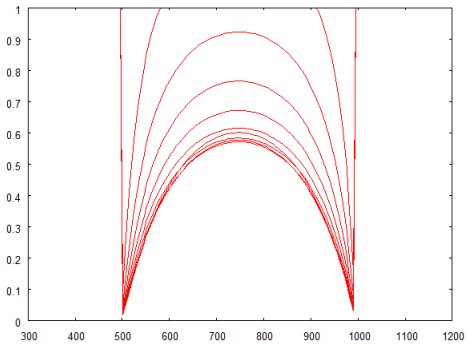
shadf.vseit



For a set of tilt angles between zero and ninety degrees as varied by a DO block in steps of ten a CLOCK block runs through the first of January 2010 from six in the morning until 7 p.m. in steps of five minutes. The SUNAE block calculates the position of the Sun for Stuttgart, Germany.

An OFFSET block with parameter -180 is used to calculate the relative surface azimuth ψ' which is required by the SHADF block. The other two inputs required by SHADF are the solar elevation and the surface tilt angle. The parameters of the SHADF block are set to a row distance of 3.65 meters at a module height of two meters.

The PLOTP block displays the unshaded fractions.



Of course, with tilt angle zero no shading occurs and the fraction of the unshaded area is constant equal to one (and not visible in the graph). With a vertical surface even at noon nearly fifty percent of the area is shaded.

2.44 Block SKYALL

The SKYALL block simulates an Perez' allsky model.

Name	SKYALL
Function	em0029
Inputs	5
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1 Total horizontal diffuse radiation [W/m^2]
2 Azimuth of the point [deg]
3 Elevation of the point [deg]
4 Azimuth of the Sun [deg]
5 Elevation of the Sun [deg]

Outputs

1 Luminance from the point [cd/m^2]
2 ?

Parameters

None

Strings

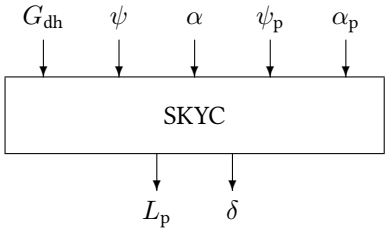
None

Remarks

The coefficients are taken from T. Muneer, Solar radiation and daylight models.

2.45 Block SKYC

The SKYC block simulates a clear sky.



Name	SKYC
Function	em0027
Inputs	5
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1 Total horizontal diffuse radiation G_{dh} / W m^{-2}
2 Solar azimuth ψ / °
3 Solar elevation α / °
4 Point azimuth ψ_p / °
5 Point elevation α_p / °

Outputs

1 Point luminance L_p / cd m^{-2}
2 Angle between point and Sun δ / °

Parameters

None

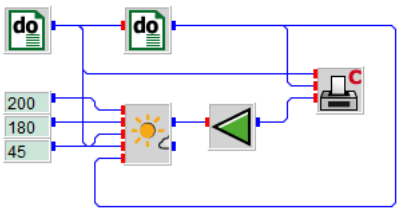
Strings

None

Remarks

The equations and coefficients are taken from the source “Daylighting in Architecture – A European Reference Book” pages 3.6 and 3.7 published for the Commission of the European Communities by James and James (Science Publishers) Ltd. 1993 Brussels and Luxembourg

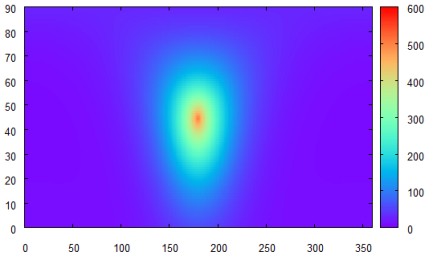
skyc.vseit



Two nested DO blocks vary elevation between zero and ninety degrees in steps of one (inner DO block) and azimuth between zero and 360 degrees in steps of 1 (outer DO block).

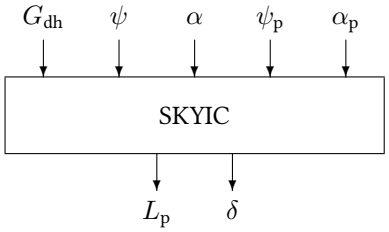
The diffuse radiation is assumed constant with 200 W/m^2. The position of the Sun is assumed at south (azimuth 180 degrees) at an altitude of 45 degrees. The luminance output of the SKYC block is attenuated by a factor 1000.

Hence, the PLOTPMC block displays the clear sky luminance distribution over the sky dome in kcd / m^2.



2.46 Block SKYIC

The SKYIC block simulates an intermediate clear sky.



Name	SKYIC
Function	em0027
Inputs	5
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1 Total horizontal diffuse radiation G_{dh} / W m^{-2}
2 Solar azimuth ψ / °
3 Solar elevation α / °
4 Point azimuth ψ_p / °
5 Point elevation α_p / °

Outputs

1 Point luminance L_p / cd m^{-2}
2 Angle between point and Sun δ / °

Parameters

None

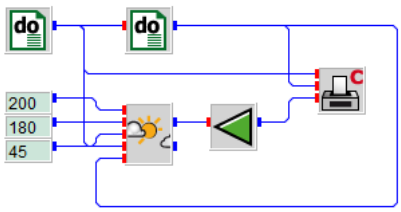
Strings

None

Remarks

The equations and coefficients are taken from the source “Daylighting in Architecture – A European Reference Book” pages 3.6 and 3.7 published for the Commission of the European Communities by James and James (Science Publishers) Ltd. 1993 Brussels and Luxembourg

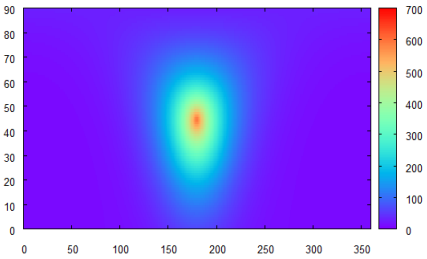
skyic.vseit



Two nested DO blocks vary elevation between zero and ninety degrees in steps of one (inner DO block) and azimuth between zero and 360 degrees in steps of 1 (outer DO block).

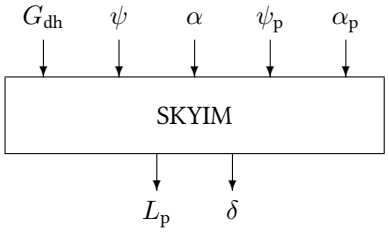
The diffuse radiation is assumed constant with 200 W/m^2. The position of the Sun is assumed at south (azimuth 180 degrees) at an altitude of 45 degrees. The luminance output of the SKYIC block is attenuated by a factor 1000.

Hence, the PLOTPMC block displays the clear sky luminance distribution over the sky dome in kcd / m^2.



2.47 Block SKYIM

The SKYIM block simulates an intermediate mean sky.



Name	SKYIM
Function	em0027
Inputs	5
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1 Total horizontal diffuse radiation G_{dh} / W m^{-2}
2 Solar azimuth ψ / °
3 Solar elevation α / °
4 Point azimuth ψ_p / °
5 Point elevation α_p / °

Outputs

1 Point luminance L_p / cd m^{-2}
2 Angle between point and Sun δ / °

Parameters

None

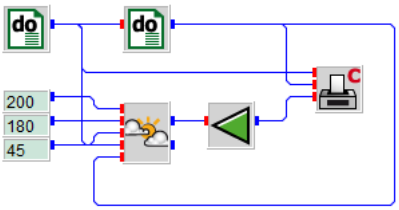
Strings

None

Remarks

The equations and coefficients are taken from the source “Daylighting in Architecture – A European Reference Book” pages 3.6 and 3.7 published for the Commission of the European Communities by James and James (Science Publishers) Ltd. 1993 Brussels and Luxembourg

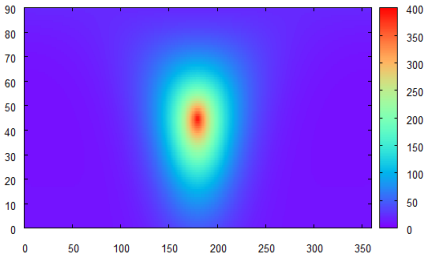
skyim.vseit



Two nested DO blocks vary elevation between zero and ninety degrees in steps of one (inner DO block) and azimuth between zero and 360 degrees in steps of 1 (outer DO block).

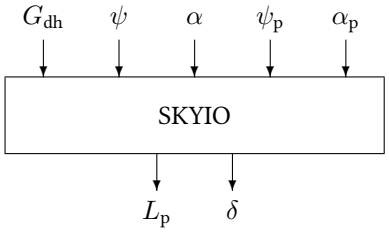
The diffuse radiation is assumed constant with 200 W/m^2. The position of the Sun is assumed at south (azimuth 180 degrees) at an altitude of 45 degrees. The luminance output of the SKYIM block is attenuated by a factor 1000.

Hence, the PLOTPMC block displays the clear sky luminance distribution over the sky dome in kcd / m^2.



2.48 Block SKYIO

The SKYIO block simulates an intermediate overcast sky.



Name	SKYIO
Function	em0027
Inputs	5
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1 Total horizontal diffuse radiation G_{dh} / W m^{-2}
2 Solar azimuth ψ / °
3 Solar elevation α / °
4 Point azimuth ψ_p / °
5 Point elevation α_p / °

Outputs

1 Point luminance L_p / cd m^{-2}
2 Angle between point and Sun δ / °

Parameters

None

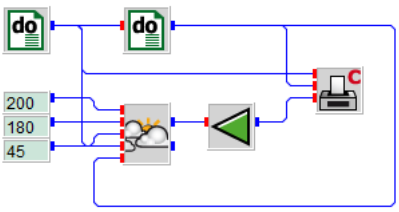
Strings

None

Remarks

The equations and coefficients are taken from the source “Daylighting in Architecture – A European Reference Book” pages 3.6 and 3.7 published for the Commission of the European Communities by James and James (Science Publishers) Ltd. 1993 Brussels and Luxembourg

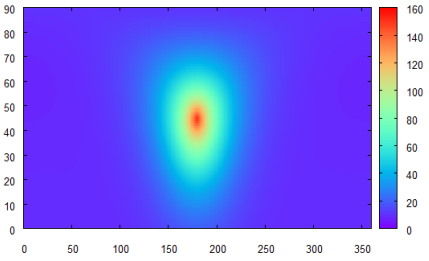
skyio.vseit



Two nested DO blocks vary elevation between zero and ninety degrees in steps of one (inner DO block) and azimuth between zero and 360 degrees in steps of 1 (outer DO block).

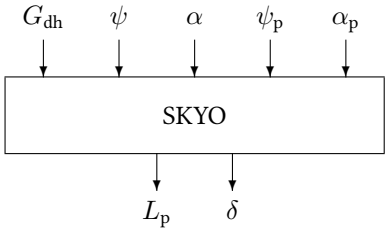
The diffuse radiation is assumed constant with 200 W/m^2. The position of the Sun is assumed at south (azimuth 180 degrees) at an altitude of 45 degrees. The luminance output of the SKYIO block is attenuated by a factor 1000.

Hence, the PLOTPMC block displays the clear sky luminance distribution over the sky dome in kcd / m^2.



2.49 Block SKYO

The SKYO block simulates an overcast sky.



Name	SKYO
Function	em0027
Inputs	5
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1 Total horizontal diffuse radiation G_{dh} / W m^{-2}
2 Solar azimuth ψ / °
3 Solar elevation α / °
4 Point azimuth ψ_p / °
5 Point elevation α_p / °

Outputs

1 Point luminance L_p / cd m^{-2}
2 Angle between point and Sun δ / °

Parameters

None

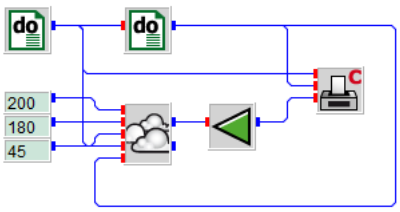
Strings

None

Remarks

The equations and coefficients are taken from the source “Daylighting in Architecture – A European Reference Book” pages 3.6 and 3.7 published for the Commission of the European Communities by James and James (Science Publishers) Ltd. 1993 Brussels and Luxembourg

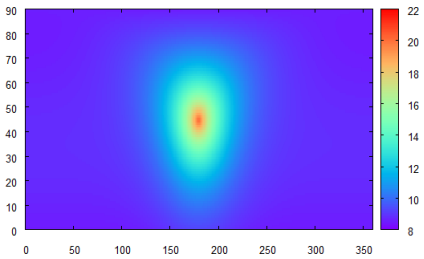
skyo.vseit



Two nested DO blocks vary elevation between zero and ninety degrees in steps of one (inner DO block) and azimuth between zero and 360 degrees in steps of 1 (outer DO block).

The diffuse radiation is assumed constant with 200 W/m^2. The position of the Sun is assumed at south (azimuth 180 degrees) at an altitude of 45 degrees. The luminance output of the SKYO block is attenuated by a factor 1000.

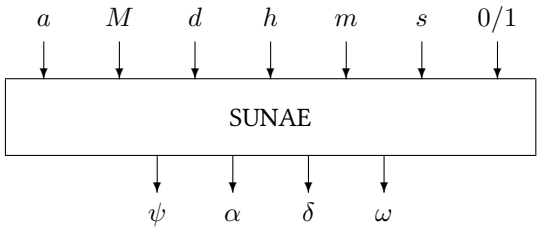
Hence, the PLOTPMC block displays the clear sky luminance distribution over the sky dome in kcd / m^2.



2.50 Block SUNAE

Warning: The explicit usage of this block is deprecated, because it uses a west-positive convention for longitudes.

Please use SUNAE2 instead, with east-positive longitudes. SUNAE is still used internally by SUNAE2. The SUNAE block calculates the position of the Sun in horizon and equator coordinates.



Model,
φ, λ, Z

Name	SUNAE
Function	em0006
Inputs	4 . . . [7]
Outputs	4
Parameters	4
Strings	0
Group	S

Inputs

1	Year a (for example 2010)
2	Month $M \in [1, 12]$
3	Day $d \in [1, 31]$
4	Hour $h \in [0, 23]$
5	Minute $m \in [0, 59]$
6	Second $s \in [0, 60)$
7	Daylight saving time (0 no, 1 yes)

Outputs

1	Azimuth ψ (north $= 0°$, east $= 90° \ldots$ west $= 270°$)
2	Elevation α / °
3	Declination δ / °
4	Hour angle ω ($0° \ldots 360°$west)

Parameters

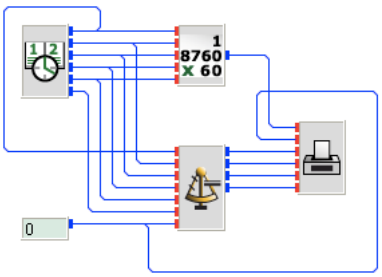
1 Approximation
  0 Spencer
  1 Holland/Mayer
  2 Michalsky
2 Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
3 Longitude $\lambda \in [0°, 360°)$, Greenwich $= 0°$, west positive
4 Time zone $Z \in [0, 23]$, e. g. Central European Time: $Z = 23$

Strings

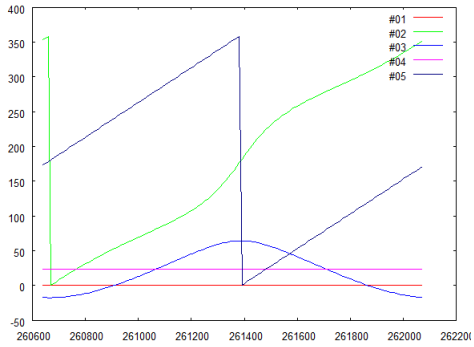
None

---

sunae.vseit



A CLOCK block varies the time of 01 July 2010 in steps of ten minutes. The SUNAE block calculates the position of the Sun for the location of Stuttgart, Germany. In order to provide a zero line a CONST block with value zero has been added. The minute-of-the-year block MOY provides a convenient scale for the x-axis of the PLOT block which displays the curves on screen.



Please notice the conventions used by INSEL:

01 Just the zero line

02 The azimuth angle is counted from 0 to 360 degrees, starting in the north with zero, clockwise positive. Consequently, when the Sun is in the South the azimuth angle is equal to $90°$, independent whether the location is on the northern orthe southern hemisphere.

03 The elevation. If positive, the Sun is above the horizon.

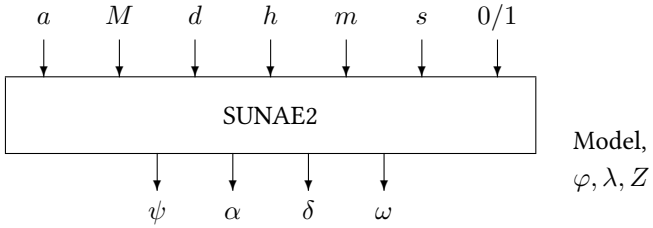
04 The declination is positive when the during a day the Sun appears in the zenith in the northern hemisphere, otherwise negative.

05 The hour angle is zero for the upper culmination of the Sun. As a consequence, true solar noon corresponds to an hour angle $\omega = 0°$

2.51 Block SUNAE2

The SUNAE2 block calculates the position of the Sun in horizon and equator coordinates.

Internally, it simply calls SUNAE which uses an old convention for timezone and longitude.



Name	SUNAE2
Function	em0106
Inputs	4 . . . [7]
Outputs	4
Parameters	4
Strings	0
Group	S

Inputs

1 Year a
2 Month $M \in [1, 12]$
3 Day $d \in [1, 31]$
4 Hour $h \in [0, 23]$
5 Minute $m \in [0, 59]$
6 Second $s \in [0, 59]$
7 Daylight saving time (0 no, 1 yes)

Outputs

1 Azimuth ψ (north $= 0°$, east $= 90° \ldots$ west $= 270°$)
2 Elevation α / °
3 Declination δ / °
4 Hour angle ω ($0° \ldots 360°$west)

Parameters

1 Approximation
0 Spencer

	1 Holland/Mayer 2 Michalsky
2	Latitude $\varphi \in [-90°, +90°]$, northern hemisphere positive
3	Longitude $\lambda \in [-180°, 180°]$, east of Greenwich; values west of Greenwich may be used with a minus sign
4	UTC Time zone $Z \in [-12, 12]$, e. g. Central European Time: $Z = +1$

Strings

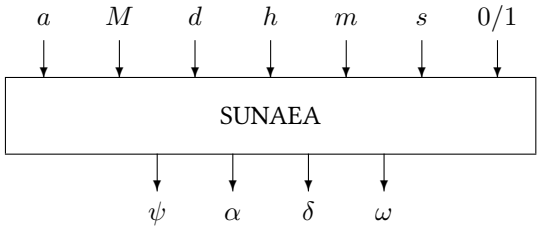
None

2.52 Block SUNAEA

Warning: The explicit usage of this block is deprecated, because it uses a west-positive convention for longitudes.

Please use SUNAEA2 instead, with east-positive longitudes. SUNAEA is still used internally by SUNAEA2.

The SUNAEA block calculates the averaged position of the Sun in horizon and equator coordinates.



Name	SUNAEA
Function	em0020
Inputs	7
Outputs	4
Parameters	4
Strings	0
Group	S

Inputs

1 Year a (for example 2002)
2 Month $M \in [1, 12]$
3 Day $d \in [1, 31]$
4 Hour $h \in [0, 23]$
5 Minute $m \in [0, 59]$
6 Second $s \in [0, 60)$
7 Daylight saving time (0 no, 1 yes)

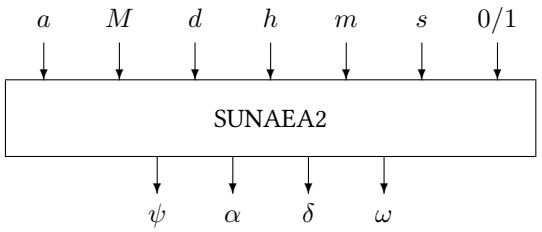
Outputs

1 Azimuth ψ (north $= 0°$, east $= 90° \ldots$ west $= 270°$)
2 Elevation α
3 Declination δ
4 Hour angle

2.53 Block SUNAEA2

The SUNAEA2 block calculates the averaged position of the Sun in horizon and equator coordinates.

Internally, it simply calls SUNAEA which uses an old convention for timezone and longitude.



Model,
φ, λ, Z

Name	SUNAEA2
Function	em0120
Inputs	7
Outputs	4
Parameters	4
Strings	0
Group	S

Inputs

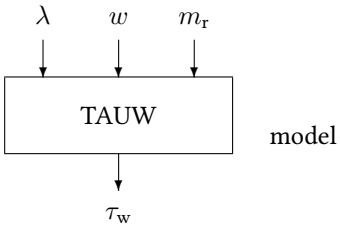
1 Year a (for example 2021)
2 Month $M \in [1, 12]$
3 Day $d \in [1, 31]$
4 Hour $h \in [0, 23]$
5 Minute $m \in [0, 59]$
6 Second $s \in [0, 60)$
7 Daylight saving time (0 no, 1 yes)

Outputs

1 Azimuth ψ (north $= 0°$, east $= 90° \ldots$ west $= 270°$)
2 Elevation α
3 Declination δ
4 Hour angle

2.54 Block TAUW

The TAUW block calculates the spectral transmittance due to water vapor in the atmosphere.



Name	TAUW
Function	em0036
Inputs	3
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Wave length λ / μm
2 Water vapor content in the atmosphere w / unit?
3 Relative air mass m_r

Outputs

1 Spectral transmittance

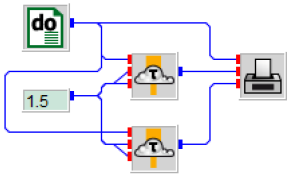
Parameters

1 Model
0 LOWTRAN
1 Lacis and Hanon

Strings

None

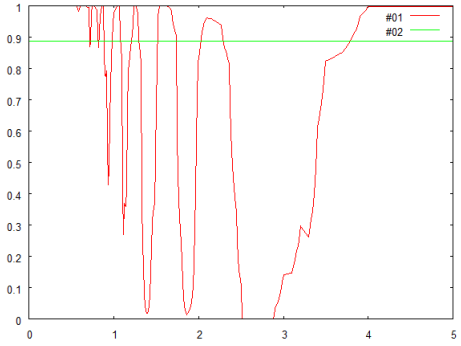
tauw.vseit



A DO block varies wavelengths between zero and five micrometers in steps of 0.01 μm. Water vapor content and relative air mass are both set constant to 1.5.

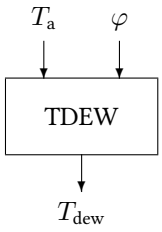
The upper TAUW block uses the wavelength dependent LOWTRAN approximation, while the lower TAUW block uses the Lacis Hanon correlation which depends on the product of water vapor and relative air mass only, and is therefore constant in this example.

As usual, the PLOT block is used to display the two curves.



2.55 Block TDEW

The TDEW block calculates the dewpoint temperature.



Name	TDEW
Function	em0034
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Ambient temperature T_a / ° C
2 Relative humidity φ

Outputs

1 Dewpoint temperature T_{dew} / ° C

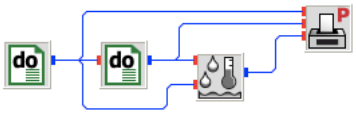
Parameters

None

Strings

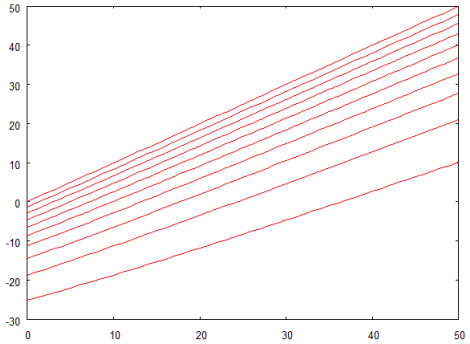
None

tdew.vseit



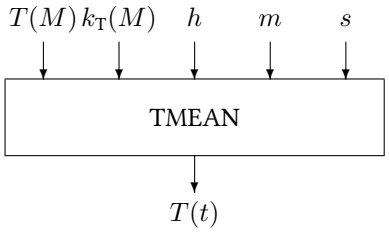
Two nested DO blocks vary the relative humidity of air between zero and one in steps of 0.1 (outer DO block) and ambient temperature between zero and fifty degrees Celsius one in steps of one (inner DO block). The data are connected to a TDEW block which calculates the dewpoint temperature.

A PLOTP block is used to display the curves.



2.56 Block TMEAN

The TMEAN block returns the ambient temperature at a given real solar time for a mean monthly ambient temperature value based on the Erbs, Klein, Beckman correlation.



Name	TMEAN
Function	em0014
Inputs	3 . . . [5]
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Monthly mean value $T(M)$ of ambient temperature / °C
2	Monthly mean value k_T of clearness index
3	Hour h
4	Minute m
5	Second s

Outputs

1	Mean ambient temperature $\bar{T}$ / °C

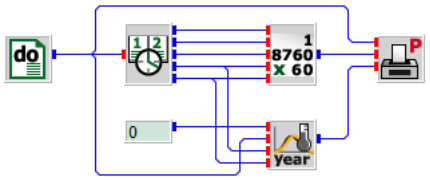
Parameters

None

Strings

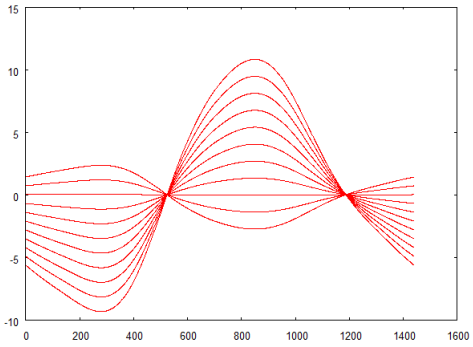
None

tmean.vseit



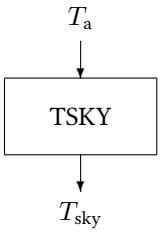
A DO block varies the monthly mean clearness index between zero and one in steps of 0.1. This signal is used by the PLOTP block as curve parameter.

A CLOCK block is used to run through the first of January 2010 in steps of one minute. The mean ambient temperature is assumed to be zero degrees Celsius. The TMEAN block calculates the mean daily temperature profile which is displayed by a PLOTP block which uses the MOY block as x-coordinate.



2.57 Block TSKY

The TSKY block calculates the sky temperature.



Model

Name	TSKY
Function	em0021
Inputs	1
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1 Ambient temperature T_a / °C

Outputs

1 Sky temperature T_{sky} / °C

Parameters

1 Model
- 0 Swinbank
- 1 Modified Swinbank

Strings

None

Description

The (effective) sky temperature T_{sky} depends on many factors such as ambient temperature T_{a}, humidity, amount and type of cloud cover, and elevation. In 1963, Swinbank proposed the equation

$$T_{\text{sky}} = 0.0552\, {T_{\text{a}}}^{1.5}$$

for clear sky conditions and averaged out other effects—both temperatures are in kelvin. However, this equation is not applicable during cloudy days, where we can expect that the sky temperature apporaches that of ambient.

In 1987, Fuentes suggests to use the average amount of cloudiness across the United States which causes approximately a 32 % decrease in the horizontal radiation and further assumes that the cloudiness and haziness causes the sky temperature to be 32 % closer to ambient on the average day than it would be during clear days. In conclusion Fuentes proposes the modified Swinbank equation

$$T_{\text{sky}} = 0.68(0.0552\, {T_{\text{a}}}^{1.5}) + 0.32 T_{\text{a}}$$

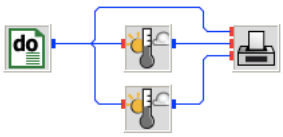
for the calculation of the sky temperature.

On the basis of extensive data from the United States Berndahl and Martin related the sky temperature to the dew point temperature T_{dp} and time

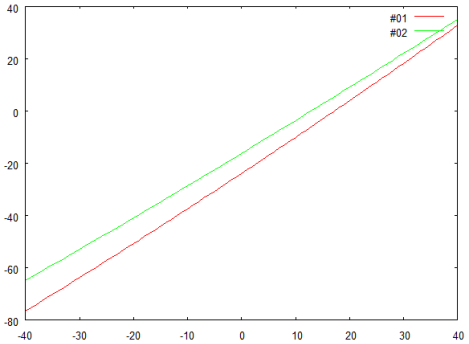
$$T_{\text{sky}} = T_{\text{a}}(0.711 + 0.0056 T_{\text{dp}} + 0.000073 T_{\text{dp}}^2 + 0.013\cos(15h))^{1/4}$$

Again, the temperatures are in kelvin, h denotes the hour of the day.

tsky.vseit

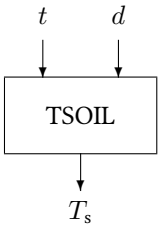


A DO block varies ambient temperature data from -40 to $+40$°C. Two TSKY blocks calculate the sky temperature according to the Swinbank equation and the modufied Swinbank equation. The PLOT block displays the graph on screen.



2.58 Block TSOIL

The TSOIL block calculates the undisturbed soil temperature according to the German VDI 4640



Name	TSOIL
Function	em0028
Inputs	2
Outputs	1
Parameters	7 . . . [10]
Strings	0 . . . [2]
Group	S

Inputs

1 Time t / s
2 Depth below ground d_T / m

Outputs

1 Undisturbed soil temperature T_{s0} / °C

Parameters

1 Input mode
  0 Read from file
  1 Enter values
  2 Select region after DIN 4710
2 Ground temperature phase delay δd / d
3 Gain factor
4 Heat capacity of the soil c_s / J kg^{-1} K^{-1}
5 Soil density ρ_s / kg m^{-3}
6 Heat conductivity of the soil λ / W m^{-1} K^{-1}
7 Number h of records to be skipped on the first call (file header, the default is $h = 0$)

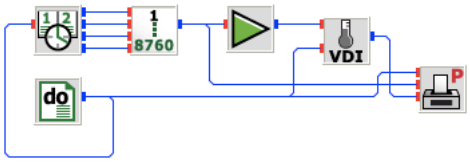
8 Annual mean temperature T_{rm} / °C
9 Maximum monthly mean temperature T_{max} / °C
10 Region select switch
- 0 Bremerhaven
- 1 Rostock
- 2 Hamburg
- 3 Potsdam
- 4 Essen
- 5 Bad Marienberg
- 6 Kassel
- 7 Braunlage
- 8 Chemnitz
- 9 Hof
- 10 Fichtelberg
- 11 Mannheim
- 12 Passau
- 13 Stötten
- 14 Garmisch-Partenkirchen

Strings

1 File name fn
2 Fortran Format

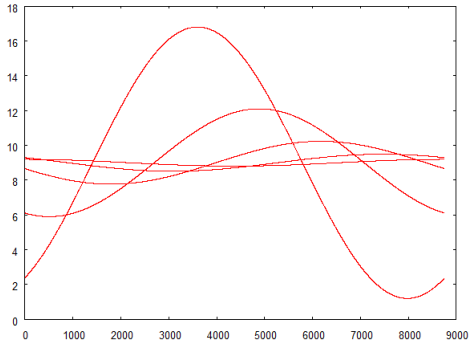
Remarks The final design of the block is not yet fixed.

tsoil.vseit



A DO block varies the depth below ground between zero and twenty meters in steps of five. The connected CLOCK block runs through the hours of the year 2010, converted to the hour of the year by the HOY block and multipled by a factor 3600 to yield the seconds of the year which are expected as input of the TSOIL block. The second input to the TSOIL block is the depth below ground which comes from the DO block.

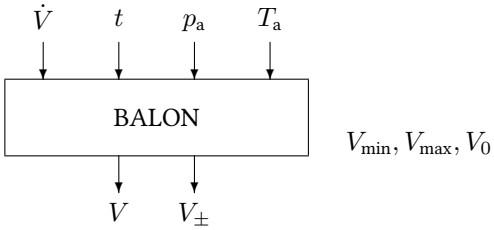
The five curves for the soil temperature are plotted by a PLOTP block.



3 :: Solar Electricity

3.1 Block BALON

The BALON block simulates a gas balloon with a savety valve.



Name	BALON
Function	se0013
Inputs	4
Outputs	2
Parameters	3
Strings	0
Group	S

Inputs

1 Volume flow $\dot{V}$ / $m_n^3\,s^{-1}$
2 Time t / s
3 Ambient pressure p_a / bar
4 Ambient temperature T_a / °C

Outputs

1 Actual volume V / m^3
2 Actual overflow/deficit $V_\pm$ / m^3

Parameters

1 Minimum volume V_{min} / m^3
2 Maximum volume V_{max} / m^3
3 Initial volume V_0 / m^3

Strings

None

Description

It is assumed that the gas is an ideal gas, the pressure in the balloon is p_a and the temparature in the balloon is T_a. Hence, the volume flow $\dot{V}$ for a time interval Δt, which is calculated from the time input, causes a change in the volume

$$V_{new} = V + \frac{(T_a + T_n)\, p_n}{T_n\, p_a} \dot{V} \Delta t$$

where

T_n	Standard temperature 273.15 K
p_n	Standard pressure 1.013 bar

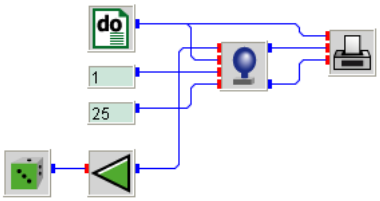
The outputs are set to

$$V = \begin{cases} V_{min} & \text{if } V_{new} < V_{min} \\ V_{new} & \text{if } V_{min} \le V_{new} \le V_{max} \\ V_{max} & \text{if } V_{new} > V_{max} \end{cases}$$

and

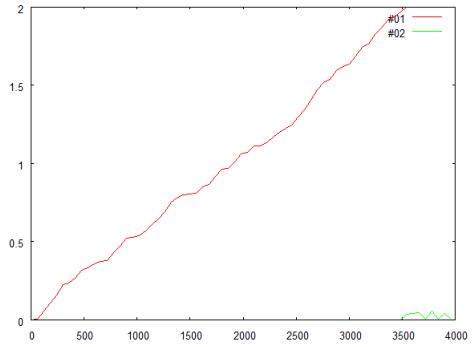
$$V_{\pm} = \begin{cases} V_{new} - V_{min} & \text{if } V_{new} < V_{min} \\ 0 & \text{if } V_{min} \le V_{new} \le V_{max} \\ V_{new} - V_{max} & \text{if } V_{new} > V_{max} \end{cases}$$

balon.vseit



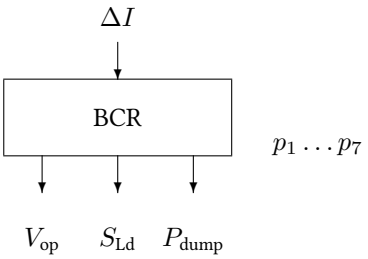
A DO block varies a time signal from zero to 4000 in steps of 60 seconds. For each time step a uniform random number is calculated by a RAN1 block and divided by a factor of 1000 by a connected ATT block. This signal is interpreted as a random volume flow in Nm^3 per second and used as input of the BALON block. The othe inputs of the BALON block are a constant pressure of one bar and an ambient temperature of 25 degrees Celsius provides by two CONST blocks.

The two outputs of the BALON block, namely the actual volume and the overflow are displayed by the PLOT block. At the end of the simulation period the balon is completely full and the are inputs handled as overflow as can be seen from the green curve.



3.2 Block BCR

The BCR block simulates a battery charge regulator. The battery voltage is kept within given limits by dumping power and switching off the load, respectively.



Name	BCR
Function	se0024
Inputs	1
Outputs	3
Parameters	7
Strings	0
Group	L

Inputs

1 Difference between the positive (energy production) and negative (load, storage) currents of the system components in the iteration loop

Outputs

1 Operating voltage V_{op} / V has to be connected to a TOL block
2 Load switch (on (1), off (0)) has to be connected to a TOL block
3 Dumped power P_{dump} / W in case of high voltage

Parameters

1 Voltage limit V_{min}^{1} / V at which the load is switched off
2 Voltage limit V_{min}^{2} / V at which the load is switched on again
3 Voltage limit V_{max} / V from where surplus power is dumped
4 Current tolerance ΔI_{max} / A for the deviation of ΔI from zero
5 Maximum number of iterations N_{max}
6 Voltage V_{fail} / V which will be set in the case when the number of iterations has reached N_{max}
7 State of the load switch S_{Ld} which will be set, when the number of iterations has reached N_{max}

Strings

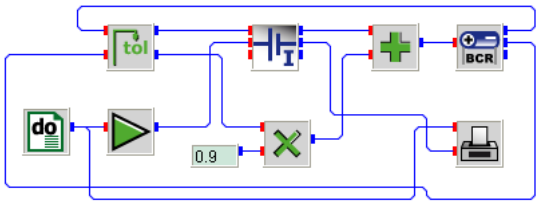
None

Description

The block keeps the operation voltage of a battery in a given range between V_{min}^{1} and V_{max}, by switching the load on/off or by dumping surplus power. This is done by varying the operation voltage U_{op} until the currents of the system components included in the loop are in balance, i.e. $|\Delta I| \leq 0 \pm \Delta I_{\text{max}}$.

The intervall between V_{min}^{1} and V_{max} is used as an iteration intervall which limits the range, in which the algorithm looks for a new output voltage V_{op}. The iteration stops when $|\Delta I| \leq \Delta I_{\text{max}}$ (or if the number of iterations reaches N_{max}). The load switch operates with an hysteresis.

bcr.vseit



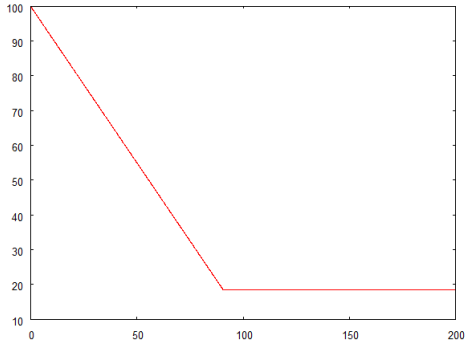
A fully charged single cell battery with a nominal capacity of 100 Ah with and simulated by a BTI block shall be dicharged at a constant current of 0.9 ampere. In order to protect the battery from deep discharge the BCR block controls the voltage range with the parameters set to a voltage limit load switch off of 2 V.

A DO block varies a time signal from zero to 200 hours in steps 0.1 hours, i. e., 6 seconds. A GAIN block with factor 1000 is used to converts the hours into a signal in seconds as required by the BTI block.

Since the BCR block requires that the sum of the input currents must finally (i. e., at the end of the iteration) be equal to zero, the SUM block adds the discharge current and the battery current. Since battery discharge current is negative by definition the load current must be taken as positive. Therefore the load current is set to 0.9 A by the CONST block and not -0.9.

When the lower battery voltage level of the battery is reached, the second output of the BCR block is set to zero. Via the TOL block this signal is multiplied into the discharge current and hence the discharge current becomes zero, i. e., the load is switched off.

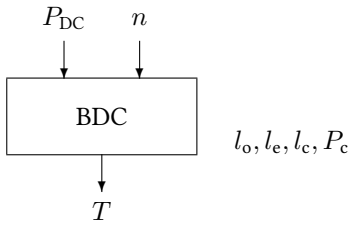
The battery capacity during the simulated period is displayed by the PLOT block.



See also Block BTI, BTV.

3.3 Block BDC

The BDC block calculates the torque of a brushless direct current motor.



Name	BDC
Function	se0007
Inputs	2
Outputs	1
Parameters	4
Strings	0
Group	S

Inputs

1 DC power P_{DC} / W
2 Rotational speed n / s^{-1}

Outputs

1 Torque T / N m

Parameters

1 Ohmic losses l_o / W N^{-1} m^{-2}
2 Electronic losses l_e / W
3 Core losses l_c / W s^2
4 Cut in power P_c / W

Strings

None

Description

According to Lenz law the torque of a motor is calculated as

$$T = c_{\mathrm{m}} I$$

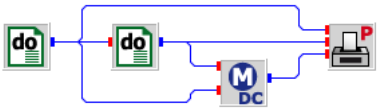
where $c_{\mathrm{m}} = dlBN$ is the motor constant with rotor diameter d, conductor length l, magnetic field density B, and N the number of conductors (2 windings each).

When loss mechanisms like ohmic (copper) losses $c_1 T^2$, constant power for the supply of the power conditioning unit c_2, and iron losses $c_3 n^2$ are considered the motor DC power can be modeled by the function

$$P_{\mathrm{DC}} = 2\pi n T + c_1 T^2 + c_2 + c_3 n^2$$

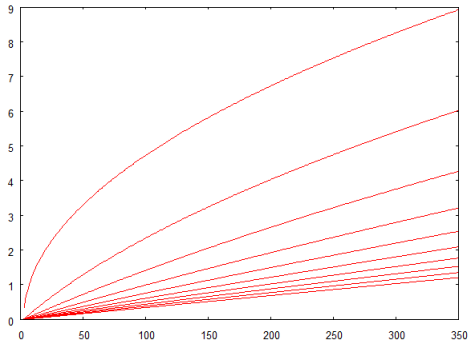
where $2\pi n T = V_{\mathrm{emf}} I$ is the mechanical power resulting from the induced back emf and the motor current.

bdc.vseit



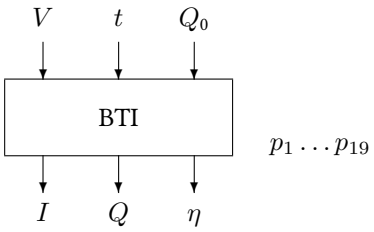
The DC power of a brushless DC motor is varied between zero and 350 watt by a DO block. The rotational speed of the motor is varied between zero and 45 per second in steps of five by a second DO block.

The torque is plotted as a function of the DC power by a PLOTP block.



3.4 Block BTI

The BTI block calculates the output current of a battery, depending on voltage and state of charge.



Name	BTI
Function	se0019
Inputs	2 . . . 3
Outputs	3
Parameters	19
Strings	0
Group	S

Inputs

1 Voltage V / V
2 Time t / s (not evaluated in mode 2)
3 Capacity Q_0 / A h (used in mode 2 only)

Outputs

1 Current I / A
2 New capacity Q / A h
3 Charging efficiency η

Parameters

1 Mode
  0 The battery capacity depends on the battery parameters and the inputs of the block; initial value of the battery capacity is given by parameter number 5 (default).
  1 The battery capacity is given by input 3.
2 Number of cells in series N_s
3 Number of cells in parallel N_p
4 Nominal capacity Q_n A h

5 Initial value of battery capacity Q_0 / A h
6 Open circuit voltage charge V_{0c} / V
7 Electrolyte coefficient (charge) g_c / V
8 Inner resistance parameter (charge) ρ_c / Ω A h
9 Battery type coefficient (charge) M_c
10 Capacity coefficient (charge) C_c
11 Open circuit voltage (discharge) V_{0d} / V
12 Electrolyte coefficient (discharge) g_d / V
13 Inner resistance parameter (discharge) ρ_d / Ω A h
14 Battery type coefficient (discharge) M_d
15 Capacity coefficient (discharge) C_d
16 Wood parameter a_w
17 Wood parameter b_w
18 Wood parameter F_ℓ
19 Self discharge coefficient q_s / % of capacity per day

Strings

None

Description

The model of the battery is based on the model of Shepherd. The relation between voltage V and current I for discharging $(I < 0)$ the battery is

$$V = V_{0d} - g_d H + \rho_d \frac{I}{Q_n}\left(1 + M_d \frac{H}{C_d - H}\right)$$

where

V_{0d} Open circuit voltage (discharge) / V
g_d Electrolyte coefficient (discharge) / V
ρ_d Inner resistance parameter (discharge) / Ω A h
M_d Battery type coeffiecient (discharge)
C_d Capacity coefficient (discharge)
H Normalised depth of discharge $1 - Q/Q_n$

For charging the following equation is used

$$V = V_{0c} - g_c H + \rho_c \frac{I}{Q_n}\left(1 + M_c \frac{F}{C_c - F}\right)$$

where

V_{0c} Open circuit voltage (charge) / V

g_c	Electrolyte coefficient (charge) / V
ρ_c	Inner resistance parameter (charge) $[V\ h]$
M_c	Battery type coffiecient (charge)
C_c	Capacity coefficient (charge)
F	Normalised state of charge Q/Q_n

In case of charging an ampere hour efficiency η after Wood/Crutcher is used. The current at which gassing effects occur is defined through

$$I_\mathrm{m} = \begin{cases} b_\mathrm{w} Q_\mathrm{n}(1-F) & \text{if } F < F_\ell \\ b_\mathrm{w} Q_\mathrm{n}(1-F_\ell) & \text{else} \end{cases}$$

where

b_w	Wood parameter
F_ℓ	Limiting state of charge / V
F	Normalized state of charge Q/Q_n
Q_n	Nominal capacity / A,h

Depending on the actual current I, the Wood-efficiency is defined as

$$\eta = \begin{cases} a_\mathrm{w} & \text{if } I \le I_\mathrm{m} \\ a_\mathrm{w} I_\mathrm{m}/I & \text{else} \end{cases}$$

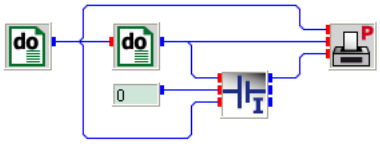
Using this efficiency, the capacity of the next timestep ($Q_{\mathrm{t}+\boxtimes\mathrm{t}}$) is calculated from

$$Q_{\mathrm{t}+\boxtimes\mathrm{t}} = Q_\mathrm{t} + \left(\eta I - \frac{q_\mathrm{s} Q}{100 t_\mathrm{s}}\right) \Delta t$$

where

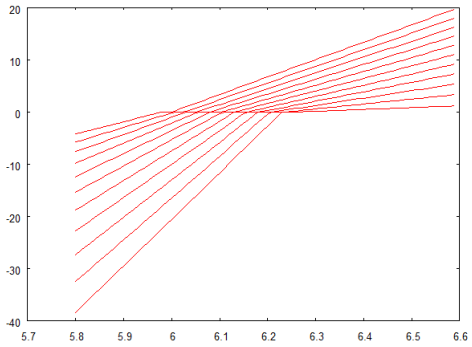
a_w	Wood parameter

bti.vseit



Two nested DO blocks vary battery capacity (outer block) and voltage (inner block) from 0 to 100 Ah in steps of 10 Ah and 5.8 to 6.6 V in steps of 0.01 V, respectively. The time input of the BTI block is set to zero by a CONST block and is of not evaluated because the capacity mode of the BTI block is set to “Battery capacity is given by input 3”.

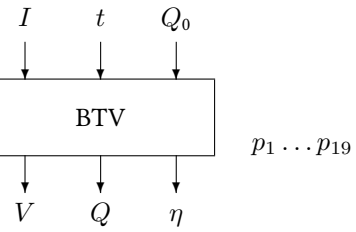
A parametric plot block PLOTP is used to display the voltage-current characteristic of the battery.



See also Block BTV, BCR.

3.5 Block BTV

The BTV block calculates the output voltage of a battery, depending on current and state of charge.



Name	BTV
Function	se0019
Inputs	2 . . . 3
Outputs	3
Parameters	19
Strings	0
Group	S

Inputs

1 Current I / A
2 Time t / s (not evaluated in mode 2)
3 Capacity Q_0 / A h (used in mode 2 only)

Outputs

1 Voltage V / V
2 New capacity Q / A h
3 Charging efficiency η

Parameters

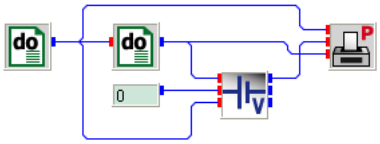
1 Mode
  0 The battery capacity depends on the battery parameters and the inputs of the block; initial value of the battery capacity is given by parameter number 5 (default).
  1 The battery capacity is given by input 3.
2 Number of cells in series N_s
3 Number of cells in parallel N_p
4 Nominal capacity Q_n A h

5 Initial value of battery capacity Q_0 / A h
6 Open circuit voltage charge V_{0c} / V
7 Electrolyte coefficient (charge) g_c / V
8 Inner resistance parameter (charge) ρ_c / Ω A h
9 Battery type coefficient (charge) M_c
10 Capacity coefficient (charge) C_c
11 Open circuit voltage (discharge) V_{0d} / V
12 Electrolyte coefficient (discharge) g_d / V
13 Inner resistance parameter (discharge) ρ_d / Ω A h
14 Battery type coefficient (discharge) M_d
15 Capacity coefficient (discharge) C_d
16 Wood parameter a_w
17 Wood parameter b_w
18 Wood parameter F_ℓ
19 Self discharge coefficient q_s / % of capacity per day

Strings

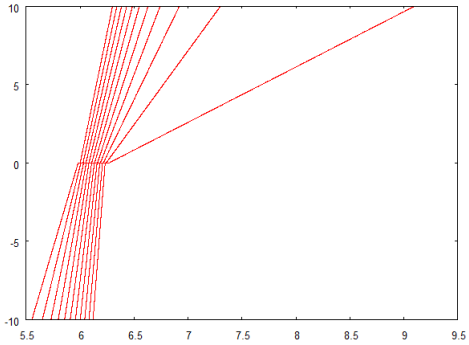
None

btv.vseit



Two nested DO blocks vary battery capacity (outer block) and current (inner block) from 0 to 100 Ah in steps of 10 Ah and -10 to 10 A in steps of 0.01 A, respectively. The time input of the BTI block is set to zero by a CONST block and is of not evaluated because the capacity mode of the BTI block is set to “Battery capacity is given by input 3”.

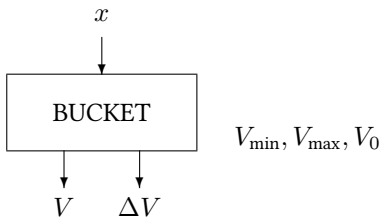
A parametric plot block PLOTP is used to display the voltage-current characteristic of the battery.



See also Block BTI, BCR.

3.6 Block BUCKET

The BUCKET block cumulates its input signal within given limits.



Name	BUCKET
Function	se0016
Inputs	1
Outputs	2
Parameters	3
Strings	0
Group	S

Inputs

1 Any signal x, which is cumulated unless the contents V reaches or exceeds a predefined threshold

Outputs

1 Actual contents of bucket V
2 Actual overflow / deficit of bucket ΔV

Parameters

1 Minimum contents $V_{\min}$
2 Maximum contents $V_{\max}$
3 Initial contents V_0

Strings

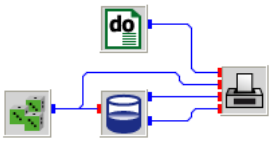
None

Description

The outputs $V_{t+\boxtimes t}$ and ΔV are calculated as

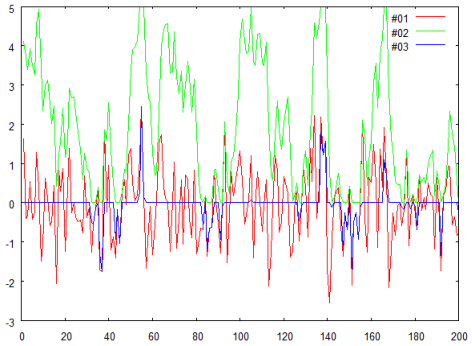
$$V_{t+\Delta t} = \begin{cases} V_{\min} & \text{if} \quad V_t \le V_{\min} \\ V_t + x; & \text{if} \quad V_t + x < V_{\max} \\ V_{max}; & \text{if} \quad V_t + x > V_{\max} \end{cases}$$

bucket.vseit



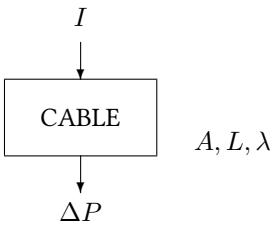
A bucket is filled with normally distributed random numbers by a GASDEV block 200 times as specified in the DO block.

The random numbers (red), contents of the bucket (green) and overflow or deficite (blue) are plotted by a PLOT block.



3.7 Block CABLE

The CABLE block calculates ohmic losses in a cable.



Name	CABLE
Function	se0003
Inputs	1
Outputs	1
Parameters	3
Strings	0
Group	S

Inputs

1 Current I / A

Outputs

1 Power loss ΔP / W

Parameters

1 Cable area A / mm^2
2 Cable length L / m (single wire)
3 Ohmic conductance λ / $\Omega\,\mathrm{mm}^2\,\mathrm{m}^{-1}$ (e. g., $\lambda_{\mathrm{Cu}} = 0.0178\,\Omega\,\mathrm{mm}^2\,\mathrm{m}^{-1}$)

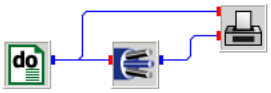
Strings

None

Decription The cable loss is calculated after

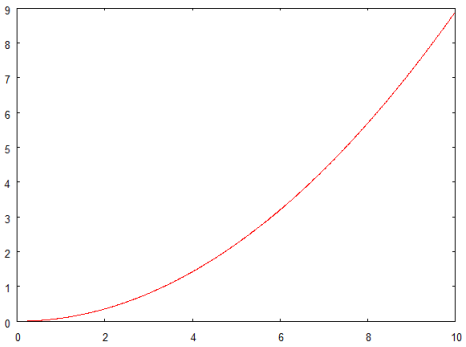
$$\Delta P = \frac{2L}{A}\,\lambda I^2$$

cable.vseit



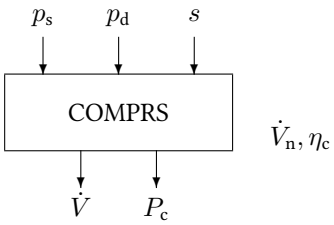
For a cable area of 4 mm^2, a length of 10 m and an ohmic conductance of 0.0178 $\Omega\,mm^2\,m^{-1}$ a DO block varies current from 0 to 10 A in steps of 0.01 A.

The PLOT block displays the ohmic losses as function of the current.



3.8 Block COMPRS

The COMPRS block simulates an isothermal compressor.



Name	COMPRS
Function	se0017
Inputs	3
Outputs	2
Parameters	2
Strings	0
Group	S

Inputs

1 Pressure at inlet cross section p_s / bar
2 Pressure at outlet cross section p_d / bar
3 Switch s on (1) off (0)

Outputs

1 Actual volume flow $\dot{V}$ / $m_n^3\,s^{-1}$
2 Power consumption of the compressor P_c / W

Parameters

1 Nominal volume flow $\dot{V}_n$ / $m_n^3\,s^{-1}$
2 Efficiency η_c

Strings

None

Description

It is assumed that the gas is an ideal gas. Hence, the power consumption of the compressor is

$$P_c = \frac{1}{\eta_c} \dot{V} p_s p_0 \ln \frac{p_d}{p_s}$$

where

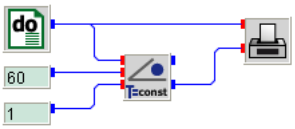
p_0 conversion factor (10^5 Pa/bar)

The outputs are set to

$$\dot{V} = \begin{cases} \dot{V}_n & \text{if } s \in [0.5, 1.5) \\ 0 & \text{else} \end{cases}$$

and P_c (or 0), respectively.

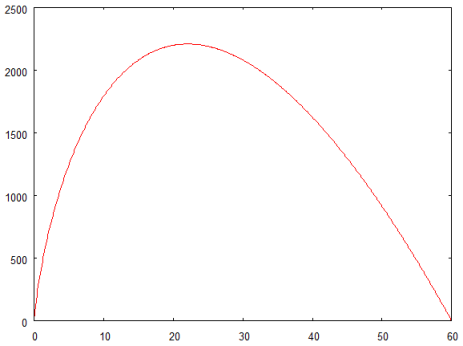
comprs.vseit



A compressor with a nominal flow of 0.5×10^{-3} m_n^3/s and an efficiency of 0.5 works against a high pressure of 60 bar, set by a CONST block.

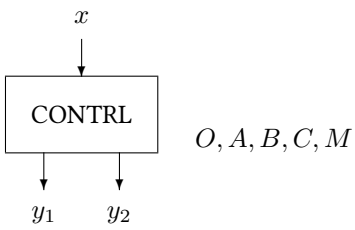
A DO block varies the inlet pressure from 0 to 60 bar is steps of 0.1 bar. Another CONST block ensures that the compressor is switched on all the time.

A PLOT block shows the resulting electrical power consumption of the compressor.



3.9 Block CONTRL

The CONTRL block simulates a switching control unit for two differnt devices.



Name	CONTRL
Function	se0014
Inputs	1
Outputs	2
Parameters	5
Strings	0
Group	D

Inputs

1	Any signal x

Outputs

1	Switch signal y_1
2	Switch signal y_2

Parameters

1	Switch level O
2	Switch level A
3	Switch level B
4	Switch level C
5	Switch level M

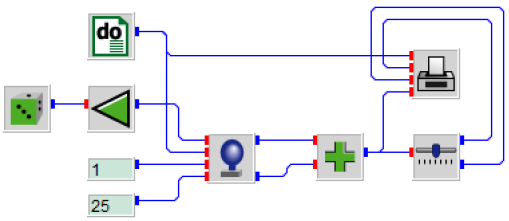
Strings

None

Description At the beginning both output signals y_1 and y_2 are set to zero. Then the outputs are set according to the following table

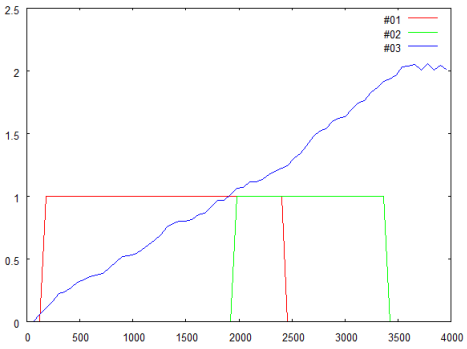
switching logics					y_1	y_2
		x	$\leq$	O	0	0
O	$<$	x	$\leq$	A	1	0
working range						
B	$\leq$	x	$<$	C	1	1
C	$\leq$	x	$<$	M	0	1
M	$\leq$	x			0	0

contrl.vseit



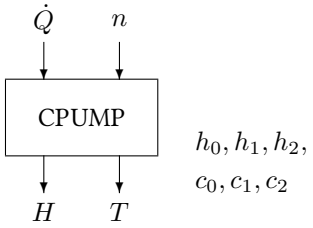
The switch levels $O = 0.05, A = 0.1, B = 1.0, C = 1.2, M = 1.9$ are used to control two devices like an elektrolysis and a fuel cell. The example is a slight adaption of the example for the BALON block.

The PLOT block displays the contents of the balloon and the two control signals. As can be seen there is a small range where both devices could work in parallel.



3.10 Block CPUMP

The CPUMP block calculates the head and torque of a centrifugal pump.



Name	CPUMP
Function	se0018
Inputs	2
Outputs	2
Parameters	6
Strings	0
Group	S

Inputs

1	Flowrate $\dot{Q}$ / $\mathrm{m^3\,h^{-1}}$
2	Rotational speed n / $\mathrm{s^{-1}}$

Outputs

1	Delivery height H / m
1	Torque T / N m

Parameters

1	Head coefficient $h_0 > 0$
2	Head coefficient $h_1 > 0$
3	Head coefficient $h_2 > 0$
4	Torque coefficient $c_0 > 0$
5	Torque coefficient $c_1 > 0$
6	Torque coefficient $c_2 > 0$

Strings

None

Description

The delivery height of a centrifugal pump is modeled by

$$H = h_0 n^2 + h_1 n\dot{Q} - h_2 \dot{Q}^2$$

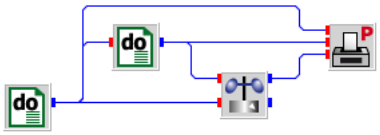
where n the rotational and $\dot{Q}$ is the flowrate. h_0, h_1, and h_2 are free positive fit parameters.

The torque is calculated by the equation

$$T = c_0 n^2 + c_1 n\dot{Q} - c_2 \dot{Q}^2$$

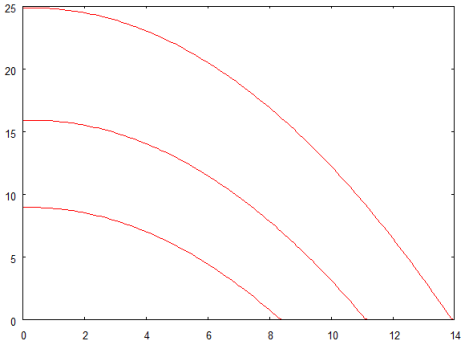
Again, c_0, c_1, and c_2 are free positive fit parameters.

cpump.vseit



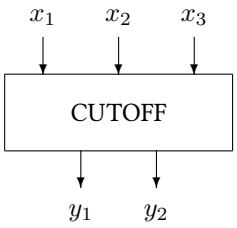
The flowrate of a centrigugal pump is varied between zero and 14 m^3 per hour by a DO block. The rotational speed of the pump is varied between 30 and 50 per second in steps of 10 by a second DO block.

The delivery height of the pump is plotted as a function of the flowrate by a PLOTP block.



3.11 Block CUTOFF

The CUTOFF block limits a signal to a restricted interval.



Name	CUTOFF
Function	se0010
Inputs	3
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1 Any signal x_1
2 Minimum value x_2 for output y_1
3 Maximum value x_3 for output y_1

Outputs

1 Restricted signal y_1, ie if $x_1 \in [x_2, x_3]$ then $y_1 = x_1$, else $y_1 = 0$,
2 Inverse restricted signal y_2, ie if $x_1 \in [x_2, x_3]$ then $y_2 = 0$, else $y_2 = x_1$,

Parameters

None

Strings

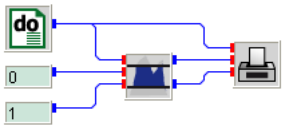
None

Description The output signals y_1 and y_2 are calculated from the inputs x_1, x_2 and x_3 according to

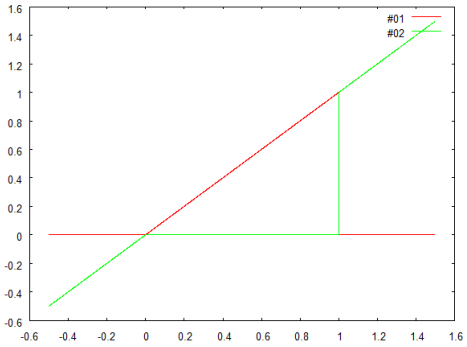
$$y_1 = \begin{cases} x_1 & \text{if } x_1 > x_2 \text{ and } x_1 < x_3 \\ 0 & \text{else} \end{cases}$$

$$y_2 = \begin{cases} 0 & \text{if } x_1 > x_2 \text{ and } x_1 < x_3 \\ x_1 & \text{else} \end{cases}$$

cutoff.vseit

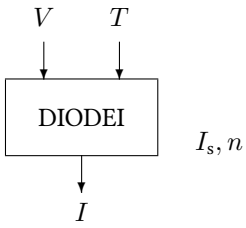


A DO block varies a signal from -0.5 to 1.5 in small steps of 0.001. The CUTOFF block restricts this signal to an interval between zero and one as displayed by the PLOT block.



3.12 Block DIODEI

The DIODEI block calculates the current of a diode as a function of voltage and temperature.



Name	DIODEI
Function	se0021
Inputs	2
Outputs	1
Parameters	2
Strings	0
Group	S

Inputs

1 Voltage V / V
2 Temperature T / °C

Outputs

1 Current I / A

Parameters

1 Saturation current I_s / A
2 Emission coefficient n

Strings

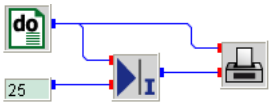
None

Description The diode current is given by

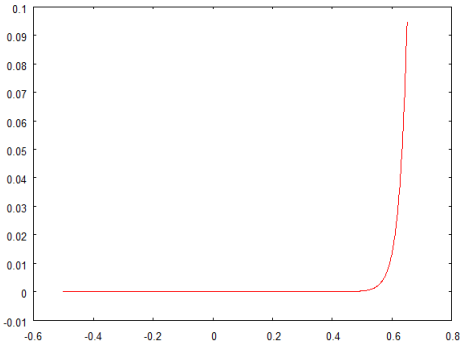
$$I = I_s \left(\exp\left(\frac{qV}{nkT} \right) - 1 \right)$$

where k is the Boltzmann constant and q the elementary charge.

diodei.vseit

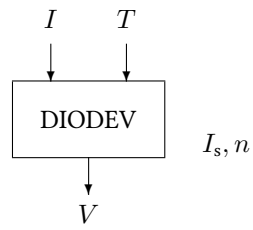


A DO block varies voltage from -0.5 to 0.65 V in steps of 0.001 V. A CONST block is used to set the diode temperature to 25 °C. For a diode saturation current of 10^{-12} A and an emission coefficient of 1 the PLOT block displays the diode's voltage-current characteristic.



3.13 Block DIODEV

The DIODEV block calculates the voltage of a diode as a function of current and temperature.



Name	DIODEV
Function	se0021
Inputs	2
Outputs	1
Parameters	2
Strings	0
Group	S

Inputs

1 Current I / A
2 Temperature T / °C

Outputs

1 Voltage V / V

Parameters

1 Saturation current I_s / A
2 Emission coefficient n

Strings

None

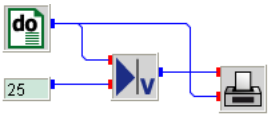
Description

The diode voltage is given by

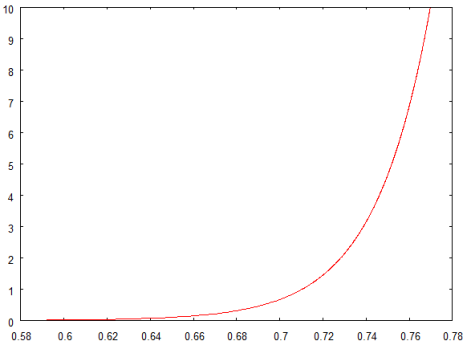
$$V = \frac{nkT}{q} \ln\left(1 + \frac{I}{I_s}\right)$$

where k is the Boltzmann constant and q the elementary charge.

diodev.vseit

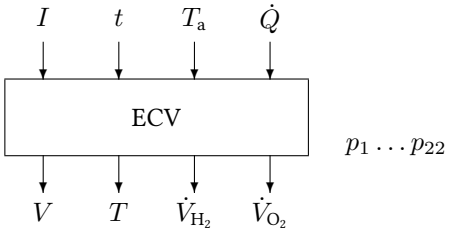


A DO block varies current from 0.01 to 10 A in steps of 0.01 A. A CONST block is used to set the diode temperature to 25 °C. For a diode saturation current of 10^{-12} A and an emission coefficient of 1 the PLOT block displays the diode's voltage-current characteristic.



3.14 Block ECV

The ECV block simulates an electrolysis cell.



Name	ECV
Function	se0011
Inputs	4
Outputs	4
Parameters	22
Strings	0
Group	S

Inputs

1 Current I / A
2 Time t / s
3 Ambient temperature T_a / °C
4 Heating / cooling power $\dot{Q}$ / W

Outputs

1 Voltage V / V
2 Cell temperature T / °C
3 Volume flow hydrogen $\dot{V}_{H_2}$ / $m_n^3\ s^{-1}$
4 Volume flow oxygen $\dot{V}_{O_2}$ / $m_n^3\ s^{-1}$

Parameters

1 Mode
  0 The cell temperature depends on the thermal parameters of the electrolysis cell and the inputs of the block; the initial value of the cell temperature is given by parameter number 22.
  1 The cell temperature is given by input 3
2 Area of a single cell A_c / m^2
3 Number of cells in series N_s

4 Number of cells in parallel N_p
5 Molar enthalpy for the reaction ΔH / J mol^{-1}
6 Equilibrium voltage U_{00} / V at reference temperature $T_0 = 298.15$ K
7 Temperature coefficient s / V K^{-1} of the equilibrium voltage
8 Number z of exchanged electrons
9 Stoichiometric factor of the negative elektrode ν_- (H_2 for water electrolysis)
10 Stoichiometric factor of the positive elektrode ν_+ (O_2 for water electrolysis)
11 Current yield factor $\epsilon_- \leq 1$ of the negative elektrode (H_2 for water electrolysis)
12 Current yield factor $\epsilon_+ \leq 1$ of the positive elektrode (O_2 for water electrolysis)
13 Exchange current densitiy coefficient C_0 / kA m^{-2}
14 Activation energy coefficient of the charge transfer C_1 / K
15 Limiting current density at reference temperature D_0 / kA m^{-2}
16 Temperature coefficient D_1 / kA m^{-2} K^{-1}
17 Conductivity coefficient at reference temperature K_0 / kΩ^{-1} m^{-2}
18 Temperature coefficient K_1 / kΩ^{-1} m^{-2} K^{-1}
19 Heat transfer coefficient K_V / W
20 Temperature exponent m
21 Heat capacity C_eff / J K^{-1}
22 Initial value of cell temperature / °C

Strings

None

Description

The model of an electrolysis cell has two parts: an electrical model and a thermal model based on an energy balance.

Electrical model

The relationship between voltage V_c of a single electrolysis cell and current density j is given by

$$V_\mathrm{c} = V_0 + \frac{4RT}{zF} \ln\left(\frac{j}{2j_\mathrm{A}} + \sqrt{\left(\frac{j}{2j_\mathrm{A}}\right)^2 + 1}\right) + \frac{RT}{zF} \ln\left(\frac{j_\mathrm{G}}{j_\mathrm{G} - j}\right) + jr_\mathrm{c}$$

where

V_0	Equilibrium voltage / V
R	Universal gas Constant 8.314 J mol^{-1} K^{-1}
z	Number of exchanged electrons
F	Faraday Constant 96 494 A s/mol
T	Cell temperature / °C

j_A Exchange current density / A m^{-2}

j_G Limiting current density / A m^{-2}

r_c Ohmic resistance coefficient / Ω m^2

This equation is valid for current densities in the range $j_A \leq j < j_G$.

The operating point of the electrolysis block is given by

$$\begin{aligned} V &= V_c N_s \\ I &= j A_c N_p \end{aligned}$$

where

N_s Total number of cells in series

N_p Total number of cells in parallel

A_c Active area of a single cell / m^2

The variables V_0, j_A, j_G and r_c depend on the cell temperature

$$\begin{aligned} V_0 &= V_{00} + s(T - T_0) \\ j_A &= C_0 \exp(-C_1/T) \\ j_G &= D_0 \exp(-D_1/T) \\ r_c &= (K_0 + K_1(T - T_0))^{-1} \end{aligned}$$

where

V_{00} Equilibrium voltage / V at reference temperature

s Temperature coefficient of equilibrium voltage / V K^{-1}

T_0 Reference temperature 298.15 K

T Cell temperature / K

C_0 Exchange current densitiy coefficient / kA m^{-2}

C_1 Activation energy coefficient of the charge transfer / K

D_0 Limiting current density at reference temperature / kA m^{-2}

D_1 Temperature coefficient / kA m^{-2} K^{-1}

K_0 Conductivity coefficient at reference temperature / kΩ m^{-2}

K_1 Temperature coefficient / kΩ m^{-2} K^{-1}

Gas yield

Only part of the current goes into the main cell reaction. Losses are described by a *current yield factor*

$$\epsilon = 1 - \frac{\sum I_{\text{side}} + I_{\text{shunt}} + I_{\text{reco}} + I_{\text{leak}}}{I}$$

where

I_{side}	Current losses due to different side reactions / A
I_{shunt}	Current losses due to ohmic shunt resistances / A
I_{reco}	Current losses due to recombinations / A
I_{leak}	Current losses due to leaks / A

Following Faradays Law $\int I \,\mathrm{d}t = nzF$, where n is the mol number, F the Faraday Constant $96\,494\,\mathrm{As/mol}$ and z is the number of exchanged charges, for the current I we have

$$V_x = \nu_x \epsilon \frac{\int I \,\mathrm{d}t}{zF} \frac{N_{\text{L}}}{N_{\text{A}}} N_{\text{s}}$$

norm cubic meter of gas x from the electrolysis block, where

ν_x	Stoichiometric coefficient of gas x ($\nu_{\text{H}_2} = 1,\ \nu_{\text{O}_2} = 1/2$),
ϵ	Current yield factor
N_{L}	Loschmidt number $6.022 \times 10^{23}\,\mathrm{mol}^{-1}$
N_{A}	Avogadro constant $2.687 \times 10^{25}\,\mathrm{m}^{-3}$
N_{s}	Number of cells in series

Thermal model

The thermal model is based on an energy balance

$$P_{\text{el}} + \dot{Q} - \dot{Q}_{\text{V}} - \Delta H \frac{I N_{\text{s}}}{zF} - C_{\text{eff}} \frac{\mathrm{d}T}{\mathrm{d}t} = 0$$

where

P_{el}	Electrical input power $U \times I$ / W
$\dot{Q}$	Heating ($\dot{Q} > 0$) or cooling ($\dot{Q} < 0$) power / W
$\dot{Q}_{\text{V}}$	Radiation and konvection losses / W
ΔH	Molar enthalpy for the reaction / $\mathrm{J\,mol}^{-1}$
C_{eff}	Heat capacity / $\mathrm{J\,K}^{-1}$]

The losses $\dot{Q}_\mathrm{V}$ are given by

$$\dot{Q}_\mathrm{V} = K_\mathrm{V} \left(\frac{T - T_\mathrm{a}}{\Delta T_0} \right)^{1+m}$$

with the heat transfer coefficient K_V / W and $\Delta T_0 = 1\,\mathrm{K}$. [1]

[1] The coefficients K_V, m and C_eff can be determined by a simple experiment: Given a constant electrolysis current I, after some time a steady state temperature T_∞ will be reached. From eq. ?? we have

$$\begin{aligned} \ln \dot{Q}_\mathrm{V} &= \ln \left(UI + \dot{Q} - \Delta H \frac{I N_\mathrm{s}}{zF} \right) \\ &= \ln K_\mathrm{V} + (1+m) \ln \left(\frac{T_\infty - T_\mathrm{a}}{\Delta T_0} \right) \end{aligned}$$

If the measurement is done with different current values I, the coefficients K_V and m can be determined by a linear regression (using block FITLIN, for example).

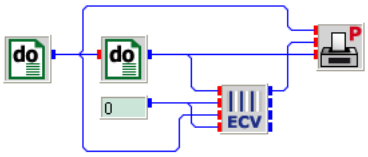
If T_∞ is reached and the current is switched off at t_off the temperature of the electrolysis block decreases till $t_\mathrm{ref} > t_\mathrm{off}$ to T_ref. Since in this case $P_\mathrm{el} = 0$ it follows for all $t > t_\mathrm{off}$ that

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{1}{C_\mathrm{eff}} \left(\dot{Q} - K_\mathrm{V} \left(\frac{T - T_\mathrm{a}}{\Delta T_0} \right)^{1+m} \right)$$

By integration from T_∞ to T_ref and the assumptions $T_\mathrm{a} = \mathrm{const}$ and $\dot{Q} = 0$ it follows

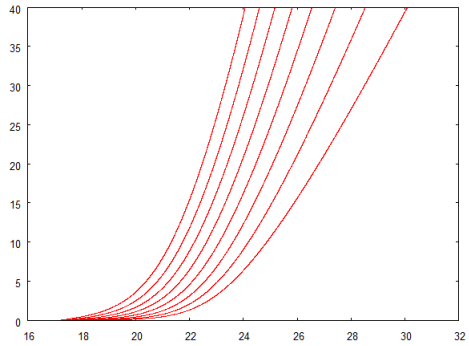
$$C_\mathrm{eff} = \frac{m K_\mathrm{V} (t_\mathrm{ref} - t_\mathrm{off})}{((T_\mathrm{ref} - T_\mathrm{a})/\Delta T_0)^{-m} - ((T_\infty - T_\mathrm{a})/\Delta T_0)^{-m}}$$

ecv.vseit



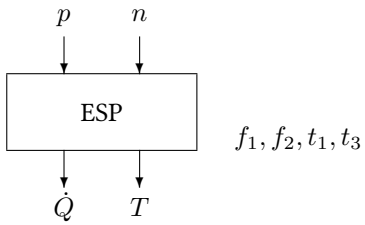
Two nested DO blocks vary temperature (outer block) and current (inner block) from 50 to 120 °C in steps of 10 °C and from 0 to 40 A in steps of 0.01 A, respectively. The time input is set to zero by a CONST block because the ECV block is operated in “Cell temperature given by input 3” mode.

A parametric PLOTP block displays the electrolyser’s voltage-current characteristics.



3.15 Block ESP

The ESP block calculates the flowrate and the torque of the excentric screw pump as a function of the delivery height and the rotational speed.



Name	ESP
Function	se0004
Inputs	2
Outputs	2
Parameters	7
Strings	0
Group	S

Inputs

1 Delivery height H / m
2 Rotational speed n / s^{-1}

Outputs

1 Flowrate $\dot{Q}$ / $m^3\,h^{-1}$
2 Required torque T / N m

Parameters

1 Displacement volume V_0 / $m^3\,s\,h^{-1}$
2 Height parameter c / m^{-2}
3 Hydraulic parameter c_0 / N
4 Friction loss T_1 / N m
5 Turbulence c_1 / $N\,m\,s^2$
6 Transition coefficient c_2 / N m
7 Transition exponent c_3 / s

Strings

None

Description

The flowrate of a screw pump (also known as progressive cavity pump or PCP) is modeled by

$$\dot{Q} = V_0 n - cH^2$$

where V_0 is the geometric volume of displacement, n the rotational speed per second, H the delivery height in meters. c is a free fit parameter.

The torque is calculated by the equation

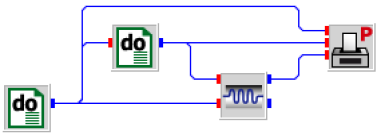
$$T = c_0 H + T_1 + c_1 n^2 + c_2 \exp(-c_3 n)$$

The first term corresponds to the hydraulic delivery rate, friction losses are assumed to cause a constant torque T_1.

$c_2 n^2$ represents the torque caused by turbulent flow in the inlet and outlet as well as within the working area in the cavities.

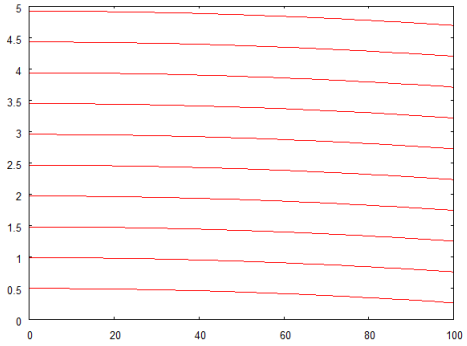
The term $c_3 \exp(-c_4 n)$ considers the torque by occasion of a transition from one form of friction to another (static friction, sliding friction, friction of laminar and turbulent flows).

esp.vseit



The delivery height of a screw pump is varied between zero and 100 meters by a DO block. The rotational speed of the pump is varied between 10 and 100 per second in steps of 10 by a second DO block.

The delivery height of the pump is plotted as a function of the flowrate by a PLOTP block.



3.16 Block ETAIS

The ETAIS block calculates the isentropic compressor efficiency.

Name	ETAIS
Function	se0036
Inputs	2
Outputs	1
Parameters	6
Strings	0
Group	S

Inputs

1 Temperature one (condensator) T_c / °C
2 Temperature two (evaporator) T_e / °C

Outputs

1 Isentropic compressor efficiency.

Parameters

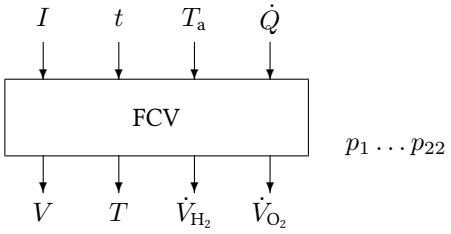
1 Coefficient a_0
2 Coefficient a_1 / K
3 Coefficient a_2 / K^2
4 Coefficient a_3 / K
5 Coefficient a_4 / K^2
6 Coefficient a_5 / K^2

Strings

None

3.17 Block FCV

The FCV block simulates a fuel cell.



Name	FCV
Function	se0012
Inputs	4
Outputs	4
Parameters	22
Strings	0
Group	S

Inputs

1 Current I / A
2 Time t / s
3 Ambient temperature T_a / °C
4 Heating /cooling power $\dot{Q}$ / W

Outputs

1 Voltage V / V
2 Cell temperature T / °C
3 Volume flow hydrogen $\dot{V}_{H_2}$ / $m_n^3\,s^{-1}$
4 Volume flow oxygen $\dot{V}_{O_2}$ / $m_n^3\,s^{-1}$

Parameters

1 Mode
  0 The cell temperature depends on the thermal parameters of the fuel cell and the inputs of the block; the initial value of the cell temperature is given by parameter number 22.
  1 The cell temperature is given by input 3.
2 Area of a single cell A_c / m^2
3 Number of cells in series N_s

4 Number of cells in parallel N_p
5 Molar enthalpy for the reaction ΔH / J mol^{-1}
6 Equilibrium voltage U_{00} / V at reference temperature $T_0 = 298.15$ K
7 Temperature coefficient s / V K^{-1} of the equilibrium voltage
8 Number z of exchanged electrons
9 Stoichiometric factor of the negative elektrode ν_- (H_2 for a H_2/O_2 fuel cell)
10 Stoichiometric factor of the positive elektrode ν_+ (O_2 for a H_2/O_2 fuel cell)
11 Current yield factor $\epsilon_- \leq 1$ of the negative elektrode (H_2 for a H_2/O_2 fuel cell)
12 Current yield factor $\epsilon_+ \leq 1$ of the positive elektrode (O_2 for a H_2/O_2 fuel cell)
13 Exchange current densitiy coefficient C_0 / kA m^{-2}
14 Activation energy coefficient of the charge transfer C_1 / K
15 Limiting current density at reference temperature D_0 / kA m^{-2}
16 Temperature coefficient D_1 / kA m^{-2} K^{-1}
17 Conductivity coefficient at reference temperature K_0 / kΩ^{-1} m^{-2}
18 Temperature coefficient K_1 / kΩ^{-1} m^{-2} K^{-1}
19 Heat transfer coefficient K_V / W
20 Temperature exponent m
21 Heat capacity C_eff / J K^{-1}
22 Initial value of cell temperature / °C

Strings

None

Description

The model of an fuel cell has two parts: an electrical model and a thermal model based on an energy balance.

Electrical model

The relationship between voltage V_c of a single fuel cell and current density j is given by

$$V_\mathrm{c} = V_0 - \frac{4RT}{zF} \ln\left(\frac{j}{2j_\mathrm{A}} + \sqrt{\left(\frac{j}{2j_\mathrm{A}}\right)^2 + 1}\right) - \frac{RT}{zF} \ln\left(\frac{j_\mathrm{G}}{j_\mathrm{G} - j}\right) - jr_\mathrm{c}$$

where

V_0	Equilibrium voltage / V
R	Universal gas Constant 8.314 J mol^{-1} K^{-1}
z	Number of exchanged electrons
F	Faraday Constant 96 494 A s/mol
T	Cell temperature / °C
j_A	Exchange current density / A m^{-2}

j_G Limiting current density / A m^{-2}

r_c Ohmic resistance coefficient / Ω m^2

This equation is valid for current densities in the range $j_A \leq j < j_G$.

The operating point of the fuel cell block is given by

$$\begin{aligned} V &= V_c N_s \\ I &= j A_c N_p \end{aligned}$$

where

N_s Total number of cells in series

N_p Total number of cells in parallel

A_c Active area of a single cell / m^2

The variables V_0, j_A, j_G and r_c depend on the cell temperature

$$\begin{aligned} V_0 &= V_{00} + s(T - T_0) \\ j_A &= C_0 \exp\left(-C_1/T\right) \\ j_G &= D_0 \exp\left(-D_1/T\right) \\ r_c &= \left(K_0 + K_1(T - T_0)\right)^{-1} \end{aligned}$$

where

V_{00} Equilibrium voltage / V at reference temperature

s Temperature coefficient of equilibrium voltage / V K^{-1}

T_0 Reference temperature 298.15 K

T Cell temperature / K

C_0 Exchange current densitiy coefficient / kA m^{-2}

C_1 Activation energy coefficient of the charge transfer / K

D_0 Limiting current density at reference temperature / kA m^{-2}

D_1 Temperature coefficient / kA m^{-2} K^{-1}

K_0 Conductivity coefficient at reference temperature / kΩ m^{-2}

K_1 Temperature coefficient / kΩ m^{-2} K^{-1}

Gas demand

Not the entire gas contributes to the generated current, losses are described by a *current yield factor*

$$\epsilon = 1 - \frac{\sum I_{\text{side}} + I_{\text{shunt}} + I_{\text{reco}} + I_{\text{leak}}}{I}$$

where

I_{side}	Current losses due to different side reactions / A
I_{shunt}	Current losses due to ohmic shunt resistances / A
I_{reco}	Current losses due to recombinations / A
I_{leak}	Current losses due to leaks / A

Following Faradays Law $\int I \,\mathrm{d}t = nzF$, where n is the mol number, F the Faraday Constant $96\,494\,\mathrm{As/mol}$ and z is the number of exchanged charges, for the current I we have

$$V_x = \nu_x \epsilon^{-1} \frac{\int I \,\mathrm{d}t}{zF} \frac{N_{\text{L}}}{N_{\text{A}}} N_{\text{s}}$$

norm cubic meter demand of gas x from the fuel cell block, where

ν_x	Stoichiometric coefficient of gas x ($\nu_{\text{H}_2} = 1,\ \nu_{\text{O}_2} = 1/2$),
ϵ	Current yield factor
N_{L}	Loschmidt Number $6.022 \times 10^{23}\,\mathrm{mol}^{-1}$
N_{A}	Avogadro Constant $2.687 \times 10^{25}\,\mathrm{m}^{-3}$
N_{s}	Number of cells in series

Thermal model

The thermal model is based on an energy balance

$$-P_{\text{el}} + \dot{Q} - \dot{Q}_{\text{V}} - \Delta H \frac{I N_{\text{s}}}{zF} - C_{\text{eff}} \frac{\mathrm{d}T}{\mathrm{d}t} = 0$$

where

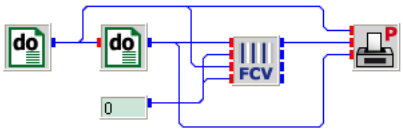
P_{el}	Electrical output power $V \times I$ / W
$\dot{Q}$	Heating ($\dot{Q} > 0$) or cooling ($\dot{Q} < 0$) power / W
$\dot{Q}_{\text{V}}$	Radiation and convection losses / W
ΔH	Molar enthalpy for the reaction / $\mathrm{J\,mol}^{-1}$
C_{eff}	Heat capacity / $\mathrm{J\,K}^{-1}$

The losses $\dot{Q}_\mathrm{V}$ are given by

$$\dot{Q}_\mathrm{V} = K_\mathrm{V} \left(\frac{T - T_\mathrm{a}}{\Delta T_0} \right)^{1+m}$$

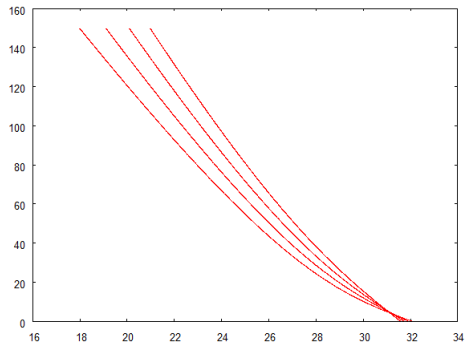
with the heat transfer coefficient K_V / W and $\Delta T_0 = 1\,\mathrm{K}$.

fcv.vseit



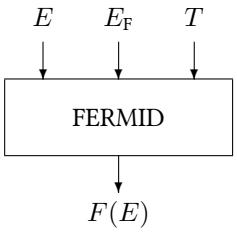
Two nested DO blocks vary temperature (outer block) and current (inner block) from 45 to 60 °C in steps of 5 °C and from 0 to 150 A in steps of 0.1 A, respectively. The time input is set to zero by a CONST block because the FCV block is operated in "Cell temperature given by input 3" mode. The heating/cooling input of the FCV block is set to zero as well.

A parametric PLOTP block displays the fuel cell's voltage-current characteristics.



3.18 Block FERMID

The FERMID block calculates the Fermi-Dirac probability that a given energy level is occupied or not.



Name	FERMID
Function	se0006
Inputs	3
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Energy E / eV
2 Fermi energy E_F / eV
3 Temperature T / K

Outputs

1 Fermi-Dirac probability that energy level E is occupied

Parameters

None

Strings

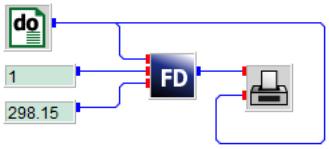
None

Description The Fermi-Dirac distribution is defined by

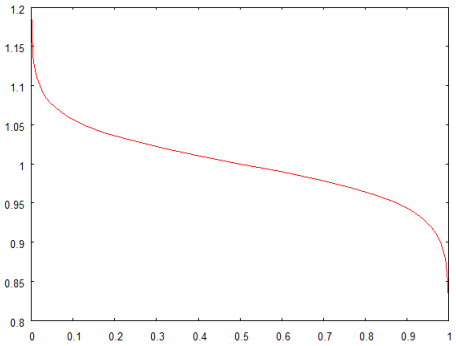
$$f(E) = \left(1 + \exp\left(\frac{E - E_{\mathrm{F}}}{kT}\right)\right)^{-1}$$

where k is the Boltzmann constant.

fermid.vseit

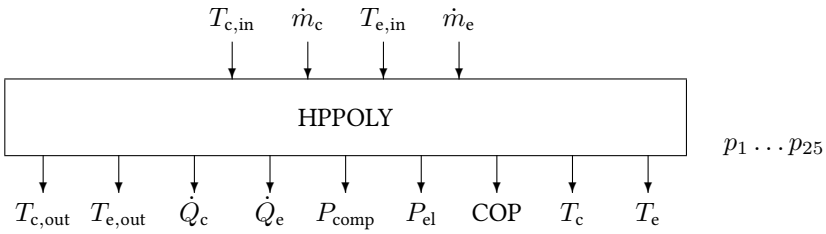


A DO block varies energy from 0.8 to 1.2 eV in steps of 0.01 eV. For a Fermi energy of 1 eV and a temperature of 298.15 K, both set by two CONST blocks the calculated Fermi-Dirac distribution is plotted by the PLOT block.



3.19 Block HPPOLY

The HPPOLY block simulates a compression heat pump by using the polynomial functions that describe the compressor performance according to EN12900.



Name	HPPOLY
Function	se0035
Inputs	4
Outputs	9
Parameters	25
Strings	0
Group	S

Inputs

1 Fluid inlet temperature at condenser side $T_{c,in}$ / °C
2 Fluid mass flow rate in the condenser $\dot{m}_c$ / kg s^{-1}
3 Fluid inlet temperature at evaporator side $T_{c,in}$ / °C
4 Fluid mass flow rate in the evaporator $\dot{m}_e$ / kg s^{-1}

Parameters

1 Number of compressors N_c
2 Specific heat capacity of fluid at evaporator side $c_{p,e}$ / kJ kg^{-1} K^{-1}
3 Specific heat capacity of fluid at condenser side $c_{p,c}$ / kJ kg^{-1} K^{-1}
4 UA value of HX at evaporator side UA_e / kW K^{-1}
5 UA value of HX at condenser side UA_c / kW K^{-1}
6 Polynomial coefficient for cooling capacity c_1
7 Polynomial coefficient for cooling capacity c_2
8 Polynomial coefficient for cooling capacity c_3
9 Polynomial coefficient for cooling capacity c_4
10 Polynomial coefficient for cooling capacity c_5
11 Polynomial coefficient for cooling capacity c_6
12 Polynomial coefficient for cooling capacity c_7

13	Polynomial coefficient for cooling capacity c_8
14	Polynomial coefficient for cooling capacity c_9
15	Polynomial coefficient for cooling capacity c_{10}
16	Polynomial coefficient for electricity consumption d_1
17	Polynomial coefficient for electricity consumption d_2
18	Polynomial coefficient for electricity consumption d_3
19	Polynomial coefficient for electricity consumption d_4
20	Polynomial coefficient for electricity consumption d_5
21	Polynomial coefficient for electricity consumption d_6
22	Polynomial coefficient for electricity consumption d_7
23	Polynomial coefficient for electricity consumption d_8
24	Polynomial coefficient for electricity consumption d_9
25	Polynomial coefficient for electricity consumption d_{10}

Outputs

1	Outlet fluid temperature at condenser side $T_{\mathrm{c,out}}$ / °C
2	Outlet fluid temperature at evaporator side $T_{\mathrm{e,out}}$ / °C
3	Power dissipated at condenser $\dot{Q}_{\mathrm{c}}$ / kW
4	Power absorbed at evaporator $\dot{Q}_{\mathrm{e}}$ / kW
5	Electrical power consumed by the compressor P_{el} / kW
6	COP of the heat pump in heating mode $\mathrm{COP_h}$
7	COP of the heat pump in cooling mode $\mathrm{COP_h}$
8	Condenser temperature T_{c} / °C
9	Evaporator temperature T_{e} / °C

Strings

None

Description

hppoly.des

3.20 Block IVETAEU

The IVETAEU block calculates the European inverter efficiency (Euro-efficiency).

Name	IVETAEU
Function	se0027
Inputs	0
Outputs	1
Parameters	6
Strings	0
Group	C

Inputs

None

Outputs

1	Euro-efficiency

Parameters

1	Inverter efficiency at 5 percent DC-nominal power
2	Inverter efficiency at 10 percent DC-nominal power
3	Inverter efficiency at 20 percent DC-nominal power
4	Inverter efficiency at 30 percent DC-nominal power
5	Inverter efficiency at 50 percent DC-nominal power
6	Inverter efficiency at 100 percent DC-nominal power

Strings

None

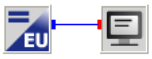
Description

The Euro-efficiency η_{EU} of an inverter weights the partial load efficiencies with average radiation conditions in Central Europe and is defined by

$$\eta_{EU} = 0.03\,\eta_5 + 0.06\,\eta_{10} + 0.13\,\eta_{20} + 0.10\,\eta_{30} + 0.48\,\eta_{50} + 0.20\,\eta_{100}$$

where η_i denotes the partial load efficiency at $i\,\%$ of the inverter's nominal DC power input.

ivetaeu.vseit



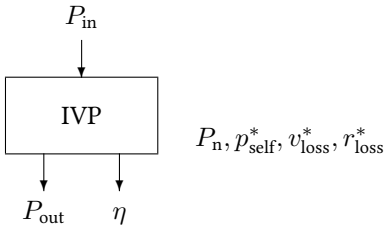
The IVETAEU block with its parameters set to $\eta_5 = 0.8$, $\eta_{10} = 0.9$, $\eta_{20} = 0.92$, $\eta_{30} = 0.93$, $\eta_{50} = 0.96$, and $\eta_{100} = 0.94$ calculates a European efficiency of

```
0.93939996
```

as displayed by the SCREEN block.

3.21 Block IVP

The IVP block simulates inverter losses.



Name	IVP
Function	se0008
Inputs	1
Outputs	2
Parameters	4
Strings	0 . . . [1]
Group	S

Inputs

1 DC power P_{in} / W

Outputs

1 AC power P_{out} / W
2 Efficiency η

Parameters

1 Nominal DC power P_n / W
2 Normalized self consumption p^*_{self}
3 Normalized voltage losses v^*_{self}
4 Normalized ohmic losses r^*_{loss}

Strings

1 Product ID

Description The model is based on a publication by Schmidt and Sauer.

$$\eta = \frac{p_{out}}{p_{in}} = \frac{p_{out}}{p_{out} + p_{loss}}$$

where $p_{out} = P_{out}/P_n$ is the fraction of the output power and the nominal power, p_{loss} is a term which describes normalized self consumption, p_{self}, normalized voltage losses v_{loss} on diodes and transistors and finally normalized ohmic losses r_{loss} following the equation

$$p_{loss} = p_{self} + v_{loss} p_{out} + r_{loss} p_{out}^2$$

which describes the efficiency as a function of output power.

When the inverter input power is used as the independent variable this leads to the equations

$$p_{loss} = p^*_{self} + v^*_{loss} p_{out} + r^*_{loss} p_{out}^2$$

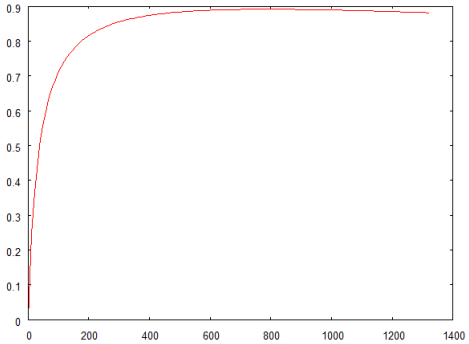
and

$$\eta = -\frac{1 + v^*_{loss}}{2 r^*_{loss} p_{in}} + \sqrt{\frac{(1 + v^*_{loss})^2}{(2 r^*_{loss} p_{in})^2} + \frac{p_{in} - p^*_{self}}{r^*_{loss} p_{in}^2}}$$

The block IVPFIT can be used to determine the required parameters from manufacturers data sheets.

ivp.vseit

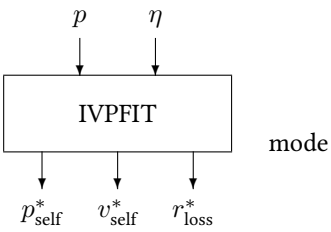
In a DO block inverter input power values are generated from 0 to 1500 W in steps of 10 W. The IVP block calculates the corresponding output power and efficiency. The PLOT block displays the efficiency as a function of the output power.



See also Block IVPFIT

3.22 Block IVPFIT

The IVPFIT calculates the parameters for the IVP block from its input data.



Name	IVPFIT
Function	se0009
Inputs	1
Outputs	3
Parameters	0 . . . [1]
Strings	0
Group	I

Inputs

1 Normalized power p
2 Efficiency η

Outputs

1 Normalized self consumption p^*_{self}
2 Normalized voltage losses v^*_{self}
3 Normalized ohmic losses r^*_{loss}

Parameters

1 Mode
- 0 Inputs are normalized with inverter input power (default)
- 1 Inputs are normalized with inverter output power

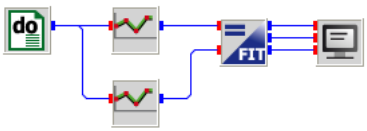
Strings

None

Remarks

The IVPFIT block uses a method suggested by Schmidt and Sauer for fitting. As a consequence, only the first three data tuples are considered in the calculation of the inverter parameters. The points should be chosen close to $p = 0.1$, $p = 0.5$ and $p = 1.0$, respectively.

ivpfit.vseit



Two POLYG blocks provide three points of the efficiency curve of an inverter. The upper block gives the three relative powers.

```
1 0.1
2 0.5
3 1.0
```

while the lower one provides the three efficiencies at 10, 50, and 100 per cent, respectively.

```
1 0.85
2 0.95
3 0.90
```

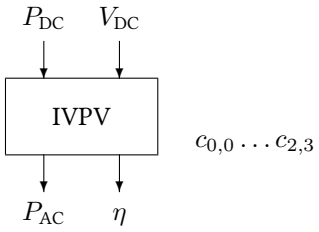
The DO block counts from 1 to 3 and the parameters calculated by the IVPFIT block are displayed by the SCREEN block.

```
 0.02029259 -0.07799642  0.18506698
```

See also Block IVP

3.23 Block IVPV

The IVPV block simulates inverter losses as a function of DC power and voltage.



Name	IVPV
Function	se0029
Inputs	2
Outputs	2
Parameters	12
Strings	0
Group	S

Inputs

1 AC power P_{AC} / W
2 DC voltage V_{DC} / V

Outputs

1 AC Power output P_{AC} / W
2 Efficiency η

Parameters

1 Fit coefficient $c_{0,0}$ / W
2 Fit coefficient $c_{0,1}$ / A
3 Fit coefficient $c_{0,2}$ / A V^{-1}
4 Fit coefficient $c_{0,3}$ / A V^{-2}
5 Fit coefficient $c_{1,0}$
6 Fit coefficient $c_{1,1}$ / V^{-1}
7 Fit coefficient $c_{1,2}$ / V^{-2}
8 Fit coefficient $c_{1,3}$ / V^{-3}
9 Fit coefficient $c_{2,0}$ / A^{-1} V^{-}1
10 Fit coefficient $c_{2,1}$ / A^{-1} V^{-}2
11 Fit coefficient $c_{2,2}$ / A^{-1} V^{-}3

12 Fit coefficient $c_{2,3}$ / $\mathrm{A}^{-1}\,\mathrm{V}^{-}4$

Strings

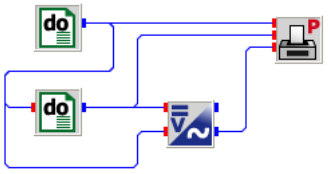
None

Description

The efficiency of the inverter is calculated from the empirical equation

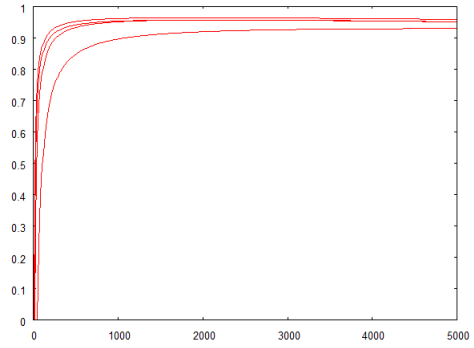
$$P_{\text{loss}} = c_{0,0} + c_{0,1}V + c_{0,2}V^2 + c_{0,3}V^3 + (c_{1,0} + c_{1,1}V + c_{1,2}V^2 + c_{1,3}V^3)P + (c_{2,0} + c_{2,1}V + c_{2,2}V^2 + c_{2,3}V^3)P^2$$

ivpv.vseit



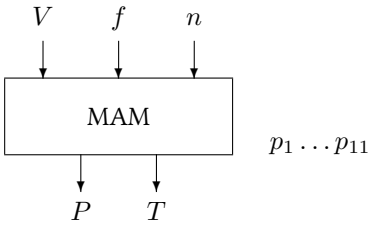
Two nested DO blocks vary voltage (outer block) from 300 to 600 V in steps of 100 V and DC power from 0 to 5000 W in steps of 10 W, respectively. The parameters of the IVPV block are taken from a publication of the 20th EUPVSEC for a SolarMax SM6000 inverter.

The voltage-dependent efficiency curves are plotted by a parametric PLOTP block.



3.24 Block MAM

The MAM block calculates the torque of an asynchronous motor.



Name	MAM
Function	se0022
Inputs	3
Outputs	1
Parameters	11
Strings	0
Group	S

Inputs

1 Voltage V / V
2 Frequency f / s^{-1}
3 Rotational speed n / s^{-1}

Outputs

1 Required electrical power P_{el} / W
2 Torque T / N m

Parameters

1 Number of pole pairs N_p
2 Number of stator phases N_s
3 Ohmic stator resistance R_s / Ω
4 Ohmic rotor resistance R_r / Ω
5 Short-circuit stator reactance $\hat{X}_s$ / Ω
6 Short-circuit rotor reactance $\hat{X}_r$ / Ω
7 Open-circuit reactance $\hat{X}_m$ / Ω
8 Nominal power core loss $\hat{P}_c$ / W
9 Open-circuit friction loss $\hat{P}_f$ / W
10 Nominal voltage V_n / V

11 Nominal frequency f_0 / s^{-1}

Strings

None

Description

The electrical power demand at the motor terminals is calculated by

$$P_{el} = N_p N_s \left(\frac{V^2}{\zeta^2} \left(\frac{R_s R_r^2}{s^2} + \frac{R_r X_m^2}{s} + R_s X_2^2 \right) + P_c \right)$$

where $s = (f - n)/n$ is the per-unit slip and ζ^2 a shortcut for

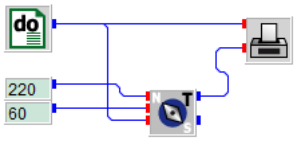
$$\zeta^2 = \left(\frac{R_s R_r}{s} + X_m^2 - X_s X_r \right)^2 + \left(\frac{R_2 X_s}{s} + R_s X_r \right)^2$$

The torque is simulated via

$$T = N_p N_s \left(\frac{R_r X_m^2 V^2}{\omega_r \zeta^2} - T_f \right)$$

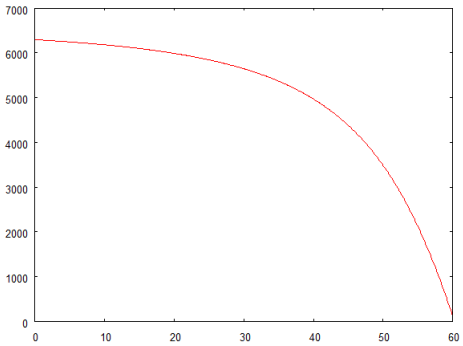
where $\omega_r = 2\pi f$ is the rotor's angular frequency.

mam.vseit



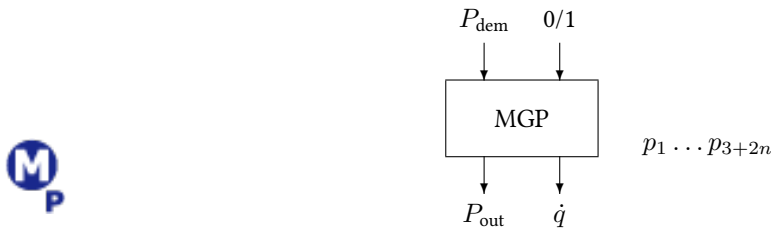
A DO block varies the rotaional speed of a motor axis between zero and 60 Hz in steps of 0.1 Hz. The voltage is fixed to 220 volt and the rotor frequency to 60 Hz by two CONST blocks.

The required electrical power is plotted by a PLOT block.



3.25 Block MGP

The MGP block returns the power output and fuel consumption of a Motor / Generator set.



Name	MGP
Function	se0020
Inputs	2
Outputs	2
Parameters	$3 + 2n$
Strings	0
Group	S

Inputs

1	Power demand P_{dem} / W
2	Switch s, which characterises the state of the motor ($s = 0$ off, $s = 1$ on)

Outputs

1	Actual power P_{out} / W of the motor
2	Actual fuel consumption $\dot{q}$ / ml s^{-1} of the motor

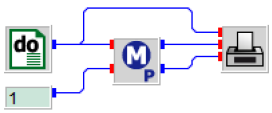
Parameters

1	Nominal power output of the motor P_N / W
2	Nominal fuel consumption $\dot{q}_n$ / ml s^{-1}
3	Number n of tabulated values of fuel consumption $y(x)$, where x represents the relative motor / generator power (normalized with the nominal power)
2i+2	Tabulated values x of normalised motor / generator power $i = \{1 \dots n\}$; the values must be given in ascending order, ie $2i + 2 < 2i + 4$ etc
2i+3	Tabulated values of fuel consumption $y(x)$ ml s^{-1}

Strings

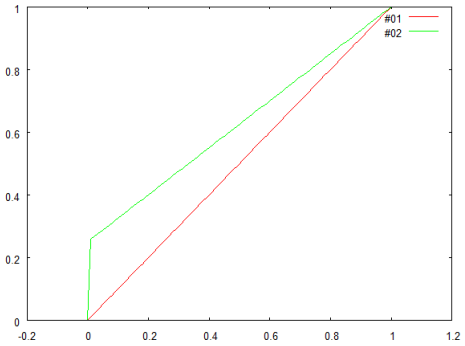
None

mgp.vseit



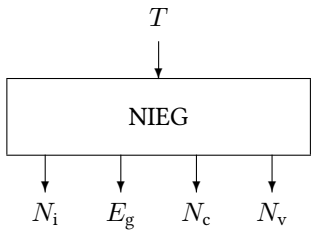
A motor normalized to a power of 1 W is simulated by an MGP block with a fuel consumption of 1 at nominal power and 0.25 at idle speed. A DO block varies the demand between −0.1 W to +1.1 W to demonstrate the behavior of the MGP block when operated out of range.

The PLOT block shows that on overload the motor power is restricted to its nominal power. For negative demand values output power and fuel consumption are set to zero.



3.26 Block NIEG

The NIEG block calculates the intrinsic charge carrier density, the energy band gap and the effective densities of states in conduction and valence bands of a semiconductor as a function of the temperature.



Name	NIEG
Function	se0031
Inputs	1
Outputs	4
Parameters	5
Strings	0
Group	S

Inputs

1	Temperature / °C

Outputs

1	Intrinsic charge carrier density / cm^{-3}
2	Band gap / eV
3	Effective density of states in the conduction band / cm^{-1}
4	Effective density of states in the valence band / cm^{-1}

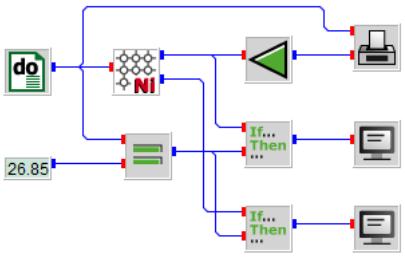
Parameters

1	Relative electon mass $m_{r,n}$
2	Relative hole mass $m_{r,p}$
3	Band gap $E_g(0)$ / eV
4	Band gap parameter a / eV K^-1
5	Band gap parameter b / K

Strings

None

nieg.vseit

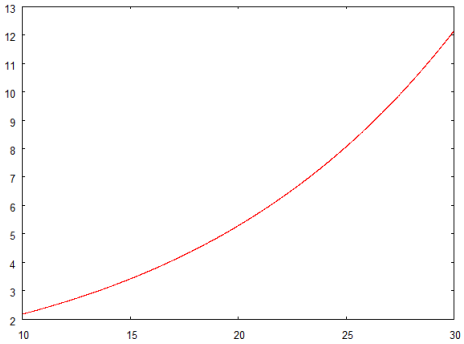


A DO block varies the temperature of a semiconductor from 10 to 30 degrees Cesius in steps of 0.01 degrees to calculate the intrinsic charge carrier density and the band gap of the material.

Two IF blocks filter the data for 26.85 degrees to display the respective values by SCREEN blocks as

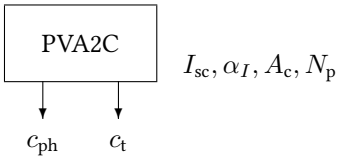
```
Intrinsic charge carrier density = 0.94067118E+10
Band gap = 1.12051928
```

A PLOT block displays the charge carrier density diluted by a factor 10^{-9} through the ATT block.



3.27 Block PVA2C

The PVA2C block calculates the photocurrent parameters of the two-diode model from the short-cicuit current at standard test conditions and its temperature coefficient.



Name	PVA2C
Function	se0033
Inputs	0
Outputs	2
Parameters	3 . . . [4]
Strings	0
Group	C

Inputs

None

Outputs

1 Coefficient c_{ph}
2 Coefficient c_t

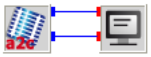
Parameters

1 Short-circuit current at STC / A
2 Temperature coefficient of short-circuit current α_I / A K^{-1}.
3 Single cell area / m^2
4 Number of cells in parallel

Strings

None

pva2c.vseit



For a short-circuit current of 8.4 A and its temperature coefficient α_I = 0.0042 A/K at standard test conditions – as usually provided on a data sheet of a PV module – the PVA2C block converts these data to the model parameters required by the PVI and PVV blocks, for instance.

The single cell area is assumed to be 0.026 m^2 with one cell in parallel. The number of cells in series has no influence on the calculations here.

The SCREEN block is used to show the results.

```
0.27491421  0.00016154
```

so that the photocurrent density is given by

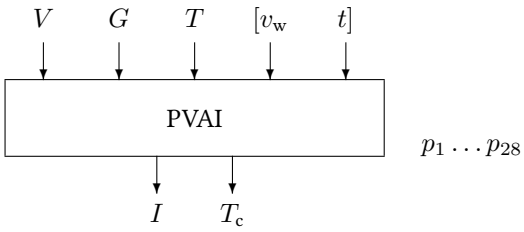
$$i_{\mathrm{ph}} = (0.27491421 + 0.00016154\,T)\;G$$

where T is the module temperature in kelvin and G the gobal irradiance in the module's plane given in W/m^2.

See also Blocks PVI, PVV, PVDET1, PVFIT2.

3.28 Block PVAI

The PVAI block calculates the output current and the temperature of a (micro-) amorphous photovoltaic generator, depending on PV generator voltage, global radiation on the generator plane, ambient temperature, and wind speed.



Name	PVAI
Function	se0028
Inputs	3 . . . [5]
Outputs	2
Parameters	28
Strings	0
Group	S

Inputs

1 Voltage V / V
2 Global radiation on the generator plane G / W m^{-2}
3 Temperature T / °C
4 Wind speed close to the generator v_w / m s^{-1}
5 Time t / s

Outputs

1 Current I / A
2 Cell temperature T_c / °C

Parameters

1 Mode
- 0 The cell temperature is given by input 3.
- 1 The cell temperature depends on the thermal parameters of the PV modules and the inputs of the block; initial value of the cell temperature is given by parameter number 26.
- 2 The difference between cell temperature and ambient temperature is a linear function of the global radiation (input 2) on the basis of NOCT.

2 Number of cells in series per module N_s
3 Number of cells in parallel N_p
4 Number of modules in series M_s
5 Number of modules in parallel M_p
6 Area of a single cell A_c / m^2
7 Area of a single module A_m / m^2
8 Coefficient of short-circuit current density c_{ph} / V^{-1}
9 Temperature coefficient of short-circuit current c_t / $V^{-1}\,K^{-1}$
10 Merten parameter M / V^{-1}
11 Built-in voltage parameter v_{bi} / V
12 Coefficient of saturation current density c_s / $A\,m^{-2}\,K^{-s}$
13 Temperature exponent s of saturation current density
14 Band gap parameter e_g / eV
15 Diode ideality factor n
16 Series resistance parameter r_s / $\Omega\,m^2$
17 Parallel resistance parameter r_p / $\Omega\,m^2$
18 Module tolerance plus / %
19 Module tolerance minus / % (< 0)
20 Characteristic module length ℓ_m / m
21 Module weight m_m / kg
22 Absorption coefficient a
23 Emission factor ϵ
24 Specific heat of a module c_m / $J\,kg^{-1}K^{-1}$
25 Nominal operating cell temperature NOCT / °C
26 Initial value of cell temperature / °C
27 Error tolerance of voltage (or current) of a single cell in the numerical iteration to solve the modified one-diode-model equation
28 Maximum number of iterations to solve the modified one-diode-model equation

Strings

None

Description

The model of the PV module has two parts: an electrical model (a modified “one-diode model”) and a thermal model based on an energy balance.

Electrical model

The relationship between voltage V_c / V of an amorphous silicon solar cell and current density j / A m^{-2} is given by the modified one-diode-model equation

$$\begin{aligned} j =& (c_{ph} + c_t T) G \left(1 - (M(v_{bi} - (V_c + j r_s)))^{-1}\right) \\ & - c_s T^s \exp\left(-\frac{e_g}{kT}\right)\left(\exp\left(\frac{q(V_c + j r_s)}{nkT}\right) - 1\right) - \frac{V_c + j r_s}{r_p} \end{aligned}$$

where the ten free fit parameters are

c_{ph}	Coefficient of short-circuit current density / V^{-1}
c_t	Temperature coefficient of short-circuit current density / V^{-1} K^{-1}
M	Merten parameter / V^{-1}
v_{bi}	Built-in voltage parameter /V
c_s	Coefficient of saturation current density / A m^{-2} K^{-s}
s	Temperature exponent of saturation current density
e_g	Band gap parameter / eV
n	Diode ideality factor
r_s	Series resistance parameter / Ω m^2
r_{sh}	Shunt resistance parameter / Ω m^2

and

T	Cell temperature / K
q	Charge of an electron (1.6021×10^{-19}As)
k	Boltzmann constant (1.3854×10^{-23}JK^{-1})

Hence, the operating point of the PV generator is given by

$$\begin{aligned} V &= V_c N_s \\ I &= j A_c N_p \end{aligned}$$

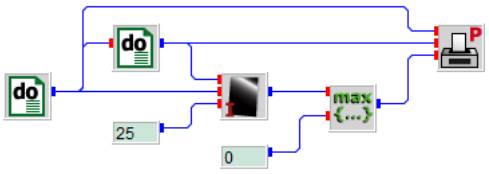
where

A_c	Area of a single cell / m^2
N_s	Number of cells in series (whole generator)
N_p	Number of cells in parallel (whole generator)

Thermal model

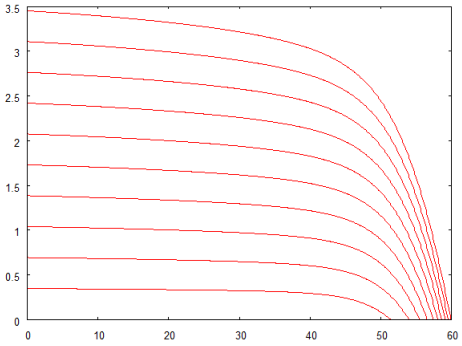
For a description of the thermal model please refer to block PVI.

pvai.vseit



Two DO blocks vary radiation between 100 and 1000 W/m^2 in steps of 100 (outer DO block) and voltage between zero and 60 V in steps of 0.1 V. The module temperature is set constant to 25 °C. These three signals are used by the PVAI block to simulate a micromorphous PV module with 180 cells in total.

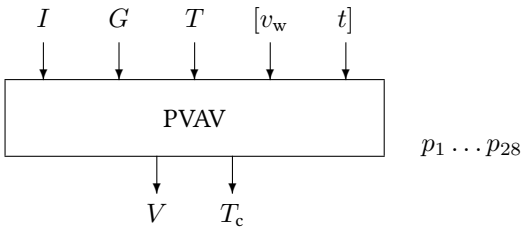
In order to avoid the output of negative currents a MAX block with a connected zero forces the PLOT block to display only values $I(V) \geq 0$.



See also Blocks PVAV, PVFITA.

3.29 Block PVAV

The PVAV block calculates the output voltage and the temperature of a (micro-) amorphous photovoltaic generator, depending on PV generator current, global radiation on the generator plane, ambient temperature, and wind speed.



Name	PVAV
Function	se0028
Inputs	3 . . . [5]
Outputs	2
Parameters	28
Strings	0
Group	S

Inputs

1 Current I / A
2 Global radiation on the generator plane G / W m^{-2}
3 Temperature T / °C
4 Wind speed close to the generator v_w / m s^{-1}
5 Time t / s

Outputs

1 Voltage V / V
2 Cell temperature T_c / °C

Parameters

1 Mode
- 0 The cell temperature is given by input 3.
- 1 The cell temperature depends on the thermal parameters of the PV modules and the inputs of the block; initial value of the cell temperature is given by parameter number 26.
- 2 The difference between cell temperature and ambient temperature is a linear function of the global radiation (input 2) on the basis of NOCT.

2 Number of cells in series per module N_s
3 Number of cells in parallel N_p
4 Number of modules in series M_s
5 Number of modules in parallel M_p
6 Area of a single cell A_c / m^2
7 Area of a single module A_m / m^2
8 Coefficient of short-circuit current density c_{ph} / V^{-1}
9 Temperature coefficient of short-circuit current c_t / $V^{-1}\,K^{-1}$
10 Merten parameter M / V^{-1}
11 Built-in voltage parameter v_{bi} / V
12 Coefficient of saturation current density c_s / $A\,m^{-2}\,K^{-s}$
13 Temperature exponent s of saturation current density
14 Band gap parameter e_g / eV
15 Diode ideality factor n
16 Series resistance parameter r_s / $\Omega\,m^2$
17 Parallel resistance parameter r_p / $\Omega\,m^2$
18 Module tolerance plus / %
19 Module tolerance minus / % (< 0)
20 Characteristic module length ℓ_m / m
21 Module weight m_m / kg
22 Absorption coefficient a
23 Emission factor ϵ
24 Specific heat of a module c_m / $J\,kg^{-1}K^{-1}$
25 Nominal operating cell temperature NOCT / °C
26 Initial value of cell temperature / °C
27 Error tolerance of voltage (or current) of a single cell in the numerical iteration to solve the modified one-diode-model equation
28 Maximum number of iterations to solve the modified one-diode-model equation

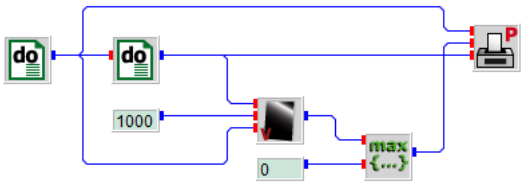
Strings

None

Description

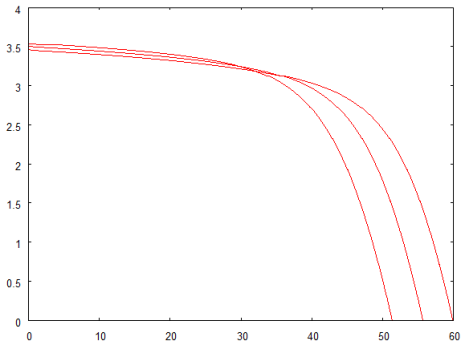
For a detailed description of the models used in the PVAV block please refer to the description of block PVAI.

pvav.vseit



Two DO blocks vary current between zero and 4 A in steps of 0.01 A and module temperature between 25 and 75 °C in steps of 25. The global radiation is set constant to 1000 W/m^2. These three signals are used by the PVAV block to simulate a micromorphous PV module with 180 cells in total.

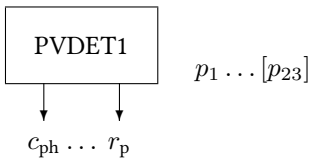
In order to avoid the output of negative voltages a MAX block with a connected zero forces the PLOT block to display only values $V(I) \geq 0$.



See also Blocks PVAI, PVFITA.

3.30 Block PVDET1

The PVDET1 block determines the one-diode-model parameters for c-Si modules from data sheet values and writes a .bp file.



Name	PVDET1
Function	se0025
Inputs	0
Outputs	6
Parameters	18 . . . [23]
Strings	6
Group	C

Inputs

None

Outputs

1 Coefficient of short-circuit current density c_{ph} / V^{-1}
2 Temperature coefficient of short-circuit current c_t / $V^{-1}\,K^{-1}$
3 Coefficient of saturation current density (Shockley diode) c_s / $A\,m^{-2}\,K^{-3}$
4 Diode ideality factor α
5 Series resistance parameter r_s / $\Omega\,m^2$
6 Parallel resistance parameter r_p / $\Omega\,m^2$

Parameters

1 Cells in series per module
2 Cells in parallel per module
3 Single cell area / m^2
4 Module area / m^2
5 Maximum power point voltage at STC V_m / V
6 Maximum power point current at STC I_m / A
7 Open-circuit voltage at STC V_{oc} / V

8 Temperature coefficient of open circuit voltage α_v / $V\,K^{-1}$
9 Short-circuit current at STC I_{sc} / A
10 Temperature coefficient of short circuit current α_i / $A\,K^{-1}$
11 Module tolerance plus / %
12 Module tolerance minus / % (≤ 0)
13 Characteristic module length / m
14 Module mass / kg
15 Absorption coefficient
16 Emission coefficient
17 Specific module heat capacity / $J\,kg^{-1}\,K^{-1}$
18 NOCT temperature / °C
19 Overwrite mode
  0 Generate error (default)
  1 Overwrite
20 Generate custom module info file record
  0 No
  1 Yes (In this case it is expected that SP(1) has the form pv9xxxxxx)
21 Maximum module voltage / V
22 Module width / m
23 Module height / m

Strings

1 INSEL module ID (pv000000 defaults to the smallest free number ≥ 900000 in file SP(6).)
2 Path to .bp file
3 Module name
4 Manufacturer
5 Cell type (e. g., cristalline)
6 Name of custom module info file

Description

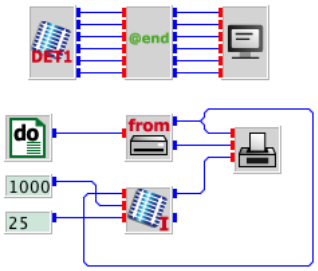
The parameters of the one- or two-diode model can be calculated uniquely from the data usually provided on the data sheets of PV module manufacturers by a rather simple set of equations. The PVDET1 block uses these equations to determine the one-diode model parameters.

pvdet1.vseit

A measured I-V curve of a typical crystalline solar cell is used to demonstrate the performance of the PVDET1 block. The data sheet of the cell would be

Nominal power P_n / W	2.3876
Nominal current I_m / A	4.8487
Nominal voltage V_m / V	0.49243
Short-circuit current I_{sc} / A	5.3125
Open-circuit voltage V_{oc} / V	0.60845
Current temperature coefficient α_I / mA K^{-1}	1.97222
Voltage temperature coefficient α_V / mV K^{-1}	−1.97222
NOCT / °C	47 ± 2
Area of a single cell A_c / m^2	0.015625
Number of cells in series N_s	1
Number of cells parallel N_p	1
Module area / m^2	0.015625
Weight / kg	0

The parameters as specified in the table above are used by the PVDET1 block.



An ATEND block is used to ensure that the SCREEN block shows the resulting values of the six block parameters only once.

```
0.27302E+00 0.22463E-03 0.59158E+04 0.10037E+01 0.12164E-03 0.37555E-01
```

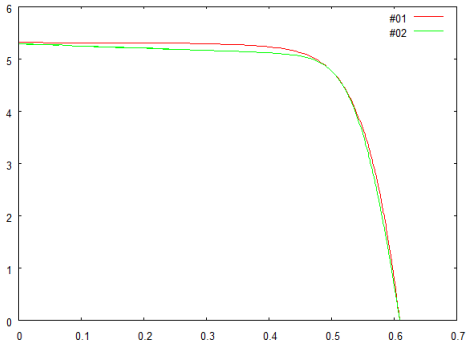
Running this INSEL model results in a file written to the directory specified in Path to .bp file. It is necessary to have write access to this directory. The file name will be as specified in the INSEL module ID parameter plus extension `.bp`, so that the example writes the file `pvdet1.bp` to the current directory.

If several new `.bp` files are created, it is recommended to use `pv900000.bp`, `pv900001.bp`, `pv900002.bp` etc and to keep track that the numbers are unique.

pvdet1.bp

```
                % Filename       pvdet1.bp
                % Module         pvdet1
                % Manufacturer   Manufacturer
                % Cell type      crystalline

                % Mode must be set externally
  1             % Number of cells in series N_s per module
  1             % Number of cells in parallel N_p per module
  1             % Number of modules in series M_s
  1             % Number of modules in parallel M_p
  0.0156        % Cell area A_c (m^2)
  0.016         % Module area A_m (m^2)

                % Electrical parameters
  1.12          % Band gap (eV)
  0.273023E+00 % Short-circuit current parameter C_0 (V^-1)
  0.224639E-03 % Isc temperature coefficient  C_1 (V^-1 K-^1)
  0.591586E+04 % Shockley saturation parameter C_01 (A m-2 K^-3)
  0.000000E+00 % Recombination saturation parameter C_02 (A m-2 K^-5/2)
  0.121641E-03 % Series resistance r_s (Ohm m^2)
  0.375550E-01 % Parallel resistance r_p (Ohm m^2)
  0.100373E+01 % Shockley diode ideality factor alpha
      2.000000 % Recombination diode quality beta
  0.000000E+00 % Bishop parameter-1
  0.000000E+00 % Bishop parameter-2
  0.000000E+00 % Bishop parameter-3
  0.0           % Module tolerance plus
  0.0           % Module tolerance minus

                % Thermal parameters
  1.000         % Characteristic module length l_m (m)
  1.000         % Module mass m_m (kg)
  0.70          % Default absorption coefficient a
  0.85          % Default emission factor epsilon
900.0           % Default specific heat of a module C_mod (J kg^-1 K^-1)
 47.0           % Nominal operating cell temperature NOCT (degrees C)
 25.0           % Intial value of cell temperature (degrees C)

                % Numerical parameters (optional)
  1E-5          % Error tolerance of voltage of single cell (V)
 100            % Maximal number of iterations to solve I/V-equation
```

The `.bp` file can directly be used in a `.insel` model. The following graph shows a comparison between the measured curve and the curve based on the obtained parameter set.



Since the open-circuit voltage, short-circuit current and maximum power point data are part of the analytical determination of the one-diode-model parameters, these points are met exactly by the method.

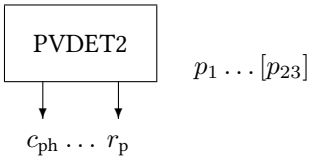
The module can be incorporated into the module data base if the option Custom module file record is set to yes. In this case a record is added to the file specified in the Name of custom module info file (incl. path) parameter.

At the moment, the only option to include the module in VSEit's PV browsers is to manually copy the new record into the sorted-by-name file `pvModules.dat` which resides in the `resources/data` directory and to copy the new `.bp` file to `resources/data/bp`.

See also Blocks PVI, PVV, PVDET2.

3.31 Block PVDET2

The PVDET2 block determines the two-diode-model parameters for c-Si modules from data sheet values and writes a .bp file.



Name	PVDET2
Function	se0025
Inputs	0
Outputs	6
Parameters	18 . . . [23]
Strings	6
Group	C

Inputs

None

Outputs

1 Coefficient of short-circuit current density c_{ph} / V^{-1}
2 Temperature coefficient of short-circuit current c_t / $V^{-1}\,K^{-1}$
3 Coefficient of saturation current density (Shockley diode) c_s / $A\,m^{-2}\,K^{-3}$
4 Coefficient of recombination current density (Recombination diode) c_r / $A\,m^{-2}\,K^{-5/2}$
5 Series resistance parameter r_s / $\Omega\,m^2$
6 Parallel resistance parameter r_p / $\Omega\,m^2$

Parameters

1 Cells in series per module
2 Cells in parallel per module
3 Single cell area / m^2
4 Module area / m^2
5 Maximum power point voltage at STC V_m / V
6 Maximum power point current at STC I_m / A

7 Open-circuit voltage at STC V_{oc} / V
8 Temperature coefficient of open circuit voltage α_v / V K^{-1}
9 Short-circuit current at STC I_{sc} / A
10 Temperature coefficient of short circuit current α_i / A K^{-1}
11 Module tolerance plus / %
12 Module tolerance minus / % (≤ 0)
13 Characteristic module length / m
14 Module mass / kg
15 Absorption coefficient
16 Emission coefficient
17 Specific module heat capacity / J kg^{-1} K^{-1}
18 NOCT temperature / °C
19 Overwrite mode
  0 Generate error (default)
  1 Overwrite
20 Generate custom module info file record
  0 No
  1 Yes (In this case it is expected that SP(1) has the form pv9xxxxxx)
21 Maximum module voltage / V
22 Module width / m
23 Module height / m

Strings

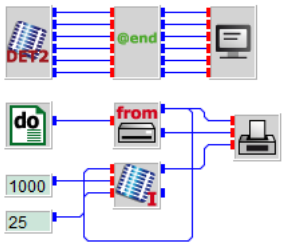
1 INSEL module ID (pv000000 defaults to the smallest free number ≥ 900000 in file SP(6).)
2 Path to .bp file
3 Module name
4 Manufacturer
5 Cell type (e. g., cristalline)
6 Name of custom module info file

Description

The parameters of the one- or two-diode model can be calculated uniquely from the data usually provided on the data sheets of PV module manufacturers by a rather simple set of equations. The PVDET2 block uses these equations to determine the two-diode model parameters.

pvdet2.vseit

A measured I-V curve of a typical crystalline solar cell is used to demonstrate the performance of the PVDET1 block. For the technical details please refer to the description of block PVDET1.



An ATEND block is used to ensure that the SCREEN block shows the resulting values of the six block parameters only once.

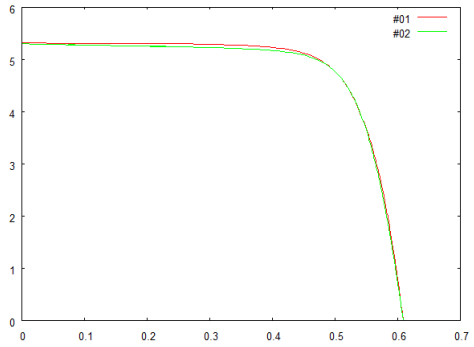
```
0.27302E+00 0.22463E-03 0.44022E+04 0.93590E+00 0.97861E-04 0.70507E-01
```

pvdet2.bp

```
                % Filename       pvdet2.bp
                % Module         pvdet2
                % Manufacturer   Manufacturer
                % Cell type      crystalline

                % Mode must be set externally
1               % Number of cells in series N_s per module
1               % Number of cells in parallel N_p per module
1               % Number of modules in series M_s
1               % Number of modules in parallel M_p
0.0156          % Cell area A_c (m^2)
0.016           % Module area A_m (m^2)

                % Electrical parameters
1.12            % Band gap (eV)
0.273023E+00 % Short-circuit current parameter C_0 (V^-1)
0.224639E-03 % Isc temperature coefficient  C_1 (V^-1 K-^1)
0.440225E+04 % Shockley saturation parameter C_01 (A m-2 K^-3)
0.935907E+00 % Recombination saturation parameter C_02 (A m-2 K^-5/2)
0.978612E-04 % Series resistance r_s (Ohm m^2)
0.705070E-01 % Parallel resistance r_p (Ohm m^2)
0.100000E+01 % Shockley diode ideality factor alpha
    2.000000 % Recombination diode quality beta
0.000000E+00 % Bishop parameter-1
0.000000E+00 % Bishop parameter-2
0.000000E+00 % Bishop parameter-3
0.0             % Module tolerance plus
0.0             % Module tolerance minus

                % Thermal parameters
1.000           % Characteristic module length l_m (m)
1.000           % Module mass m_m (kg)
```

```
  0.70          % Default absorption coefficient a
  0.85          % Default emission factor epsilon
900.0           % Default specific heat of a module C_mod (J kg^-1 K^-1)
 47.0           % Nominal operating cell temperature NOCT (degrees C)
 25.0           % Intial value of cell temperature (degrees C)

                % Numerical parameters (optional)
  1E-5          % Error tolerance of voltage of single cell (V)
100             % Maximal number of iterations to solve I/V-equation
```

The `.bp` file can directly be used in a `.insel` model. The following graph shows a comparison between the measured curve and the curve based on the obtained parameter set.

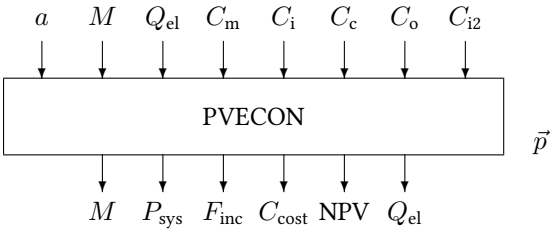


Since the open-circuit voltage, short-circuit current and maximum power point data are part of the analytical determination of the two-diode-model parameters, these points are met exactly by the method.

See also Blocks PVI, PVV, PVDET1.

3.32 Block PVECON

The PVECON block calculates the cost annuity of a grid-connected PV generator.



Name	PVECON
Function	se0030
Inputs	9
Outputs	6
Parameters	19
Strings	0
Group	T

Inputs

1 Year
2 Month of the year
3 Produced electrical energy per month Q_{el} / kWh per month
4 Costs of installed modules / Euro
5 Costs of installed inverters / Euro
6 Costs of installed cables / Euro
7 Other system costs / Euro
8 Installation costs / Euro
9 Annual module degradation / percent

Outputs

1 Month of the year
2 Credit rest value P_{sys} / Euro
3 Monthly income F_{inc} / Euro per month
4 Monthly costs C_{cost} / Euro per month
5 Net present value NPV / Euro
6 Produced electrical energy per month Q_{el} / kWh per month
7 Total electricity price / Euro per kWh

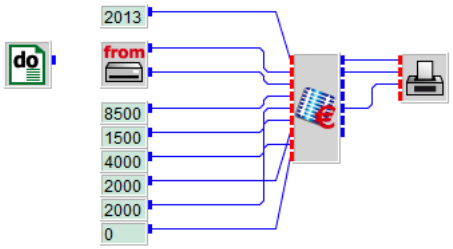
Parameters

1	Hourly rate of installer for inverter exchange $P_{\text{ex,inv}}$ / Euro per hour
2	Installation time inverter exchange $t_{\text{ex,inv}}$ / hours
3	Lifetime of inverters t_{life} /years
4	Mean time to repair module / hours
5	Hourly rate for maintenance $P_{\text{maint,h}}$ / Euro per hour
6	Mean time between module failures / years
7	Recycling costs for all installed inverters $P_{\text{recy,in}}$ / Euro
8	Recycling costs of the whole PV system including deinstallation $P_{\text{recy,PVSyst}}$ / Euro
9	Credit interest rate (lending rate) d / percent
10	Inflation rate d / percent
11	Feed in tarif F_1 / Euro per kWh
12	Lifetime of the PV modules N / years
13	Term of special feed in tariff t / years
14	Feed in tariff after term of special tariff F_2 / Euro per kWh
15	Market discount rate (account credit interest) / percent
16	Mean time to repair inverter / hours
17	Mean time between inverter failures / years
18	Maintenance time to repair inverter failure / hours
19	Maintenance time to repair module failure / hours

Strings

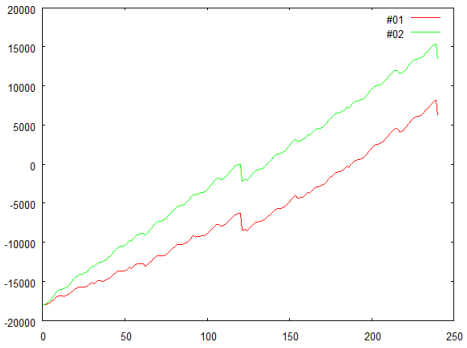
None

pvecon.vseit



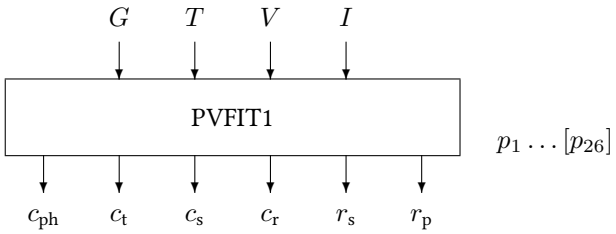
A file named `pvecon.dat` located in the `examples/data` directory is read by a READ block. The file contains twelve monthly yield data of a simulated PV generator. A DO block ensures that the twelve values are read. All other inputs of the PVECON block are set constant.

A PLOT block displays twenty years of monthly data for the credit rest value and the net present value in Euro.



3.33 Block PVFIT1

The PVFIT1 block performs a one-diode-model fit on I-V curves of a PV module connected to the block's inputs.



Name	PVFIT1
Function	se0034
Inputs	4
Outputs	7
Parameters	25 . . . [26]
Strings	5
Group	I

Inputs

1 Global radiation / $\mathrm{W\,m^{-2}}$
2 Module temperature / °C
3 Module voltage / V
4 Module current / A

Outputs

1 Coefficient of short-circuit current density c_{ph} / $\mathrm{V^{-1}}$
2 Temperature coefficient of short-circuit current c_{t} / $\mathrm{V^{-1}\,K^{-1}}$
3 Coefficient of saturation current density (Shockley diode) c_{s} / $\mathrm{A\,m^{-2}\,K^{-3}}$
4 Diode ideality factor α
5 Series resistance parameter r_{s} / $\Omega\,\mathrm{m^2}$
6 Parallel resistance parameter r_{p} / $\Omega\,\mathrm{m^2}$
7 Deviation χ

Parameters

1 Fit procedure
  0 Nelder Mead downhill simplex method
  1 Marquard Levenberg algorithm (not yet implemented)

2 Number of cells in series
3 Number of cells in parallel
4 Area of a single cell / m^2
5 Module area / m^2
6 Minimum photocurrent parameter / V^{-1}
7 Maximum photocurrent parameter / V^{-1}
8 Minimum temperature coefficient / $V^{-1} K^{-1}$
9 Maximum temperature coefficient / $V^{-1} K^{-1}$
10 Minimum saturation-current parameter
11 Maximum saturation-current parameter
12 Minimum diode-quality parameter
13 Maximum diode-quality parameter
14 Minimum series resistance / Ωm^2
15 Maximum series resistance / Ωm^2
16 Minimum parallel resistance / Ωm^2
17 Maximum parallel resistance / Ωm^2
18 Module tolerance plus / $\% > 0$
19 Module tolerance minus / $\% < 0$
20 Characteristic module length / m
21 Module weight / kg
22 Absorption coefficient a
23 Emission coefficient ϵ
24 Specific module heat capacity / $J kg^{-1} K^{-1}$
25 NOCT temperature / °C
26 Overwrite mode
  0 Generate error (default)
  1 Overwrite

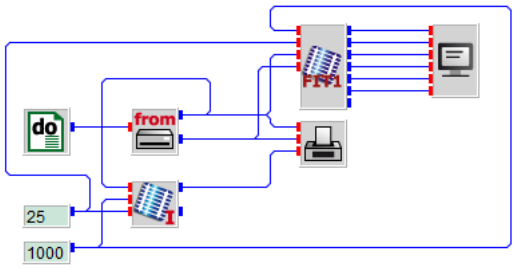
Strings

1 INSEL module ID (max. eight characters)
2 Path to .bp file
3 Module name
4 Manufacturer
5 Cell type (e. g., monocrystalline)

Remarks

The PVFIT1 block performs a fit to given data points provided by the inputs. It is not possible to determine the Bishop parameters with the PVFIT1 block.

pvfit1.vseit



A file with measured I-V curve data named `pvcell.dat`, located in the `examples/data` directory for standard test conditions is read by a READ block driven by a DO block. For details of the technical data of the cell please refer to the description of block PFDET1. The radiation and module temperature are set by two CONST blocks to 1000 W/m^2 and 25 °C, respectively.

The tupels (G, T, V, I) are fed to the PVFIT1 block which performs the parameter fit. As a result a file named `pvfit1.bp` is generated and the results are displayed by a SCREEN block.

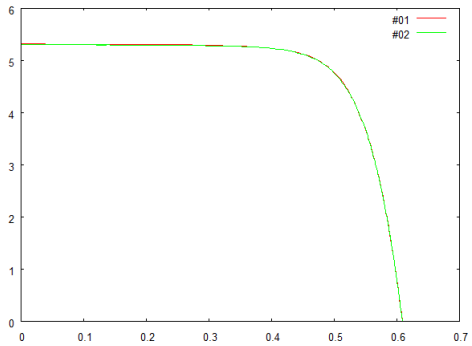
```
0.26730E+00 0.24355E-03 0.54241E+08 0.16316E+01 0.38628E-04 0.27776E+00
```

pvfit1.bp

```
                % Filename       pvfit1.bp
                % Module         pvfit1
                % Manufacturer   Manufacturer
                % Cell type      crystalline

                % Mode must be set externally
1               % Number of cells in series N_s per module
1               % Number of cells in parallel N_p per module
1               % Number of modules in series M_s
1               % Number of modules in parallel M_p
0.0156          % Cell area A_c (m^2)
0.016           % Module area A_m (m^2)

                % Electrical parameters
1.12            % Band gap (eV)
0.267307E+00 % Short-circuit current parameter C_0 (V^-1)
0.243551E-03 % Isc temperature coefficient  C_1 (V^-1 K-^1)
0.542419E+08 % Shockley saturation parameter C_01 (A m-2 K^-3)
0.000000E+00 % Recombination saturation parameter C_02 (A m-2 K^-5/2)
0.386289E-04 % Series resistance r_s (Ohm m^2)
0.277769E+00 % Parallel resistance r_p (Ohm m^2)
0.163165E+01 % Shockley diode ideality factor alpha
    2.000000 % Recombination diode quality beta
```

```
  0.000000E+00 % Bishop parameter-1
  0.000000E+00 % Bishop parameter-2
  0.000000E+00 % Bishop parameter-3
  5.0          % Module tolerance plus
 -5.0          % Module tolerance minus

               % Thermal parameters
  1.409        % Characteristic module length l_m (m)
 19.000        % Module mass m_m (kg)
  0.70         % Default absorption coefficient a
  0.85         % Default emission factor epsilon
900.0          % Default specific heat of a module C_mod (J kg^-1 K^-1)
  1.0          % Nominal operating cell temperature NOCT (degrees C)
 25.0          % Intial value of cell temperature (degrees C)

               % Numerical parameters (optional)
  1E-5         % Error tolerance of voltage of single cell (V)
100            % Maximal number of iterations to solve I/V-equation
```

The parameters have been incorporated into the `pvfit1.vseit` example using a PVI block in order to demonstrate the quality of the fit. The PLOT block then shows the measured data (red curve) and the simulated points (green curve).



Temperature dependency

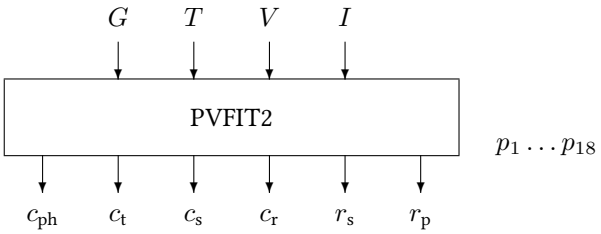
Please note that there is no guarantee that the temperature dependency of the cell is modeled correctly if only one curve is provided to the fit procedure. The most critical parameters are c_{ph} and c_{t}. Their search intervals can be narrowed if the values are known from data sheet information, for instance.

See also

Blocks PVI, PVV, PVFIT2.

3.34 Block PVFIT2

The PVFIT2 block performs a two-diode-model fit on I-V curves of a PV module connected to the block's inputs.



Name	PVFIT2
Function	se0026
Inputs	4
Outputs	7
Parameters	18
Strings	6
Group	I

Inputs

1 Global radiation / $\mathrm{W\,m^{-2}}$
2 Module temperature / °
3 Module voltage / V
4 Module current / A

Outputs

1 Coefficient of short-circuit current density c_{ph} / $\mathrm{V^{-1}}$
2 Temperature coefficient of short-circuit current c_t / $\mathrm{V^{-1}\,K^{-1}}$
3 Coefficient of saturation current density (Shockley diode) c_s / $\mathrm{A\,m^{-2}\,K^{-3}}$
4 Coefficient of recombination current density (Recombination diode) c_r / $\mathrm{A\,m^{-2}\,K^{-5/2}}$
5 Series resistance parameter r_s / $\Omega\,\mathrm{m^2}$
6 Parallel resistance parameter r_p / $\Omega\,\mathrm{m^2}$
7 Deviation χ

Parameters

1 Fit procedure
  0 Christian Obst

2 Number of cells in series
3 Number of cells in parallel
4 Area of a single cell / m^2
5 Module area / m^2
6 $R_{s,min}$ / $\Omega\,m^2$
7 $R_{s,max}$ / $\Omega\,m^2$
8 Coefficient of short-circuit current density C_0 / V^{-1}
9 Temperature coefficient of short-circuit current C_1 / $V^{-1}\,K^{-1}$
10 Module tolerance plus / %
11 Module tolerance minus / % (< 0)
12 Characteristic module length / m
13 Module mass / kg
14 Absorption coefficient
15 Emission coefficient
16 Specific module heat capacity / $J\,kg^{-1}\,K^{-1}$
17 NOCT temperature / °C
18 Overwrite mode
  0 Generate error (default)
  1 Overwrite

Strings

1 INSEL module ID (max. eight characters)
2 Path to .bp file
3 Module name
4 Manufacturer
5 Cell type (e. g., cristalline)

Remarks

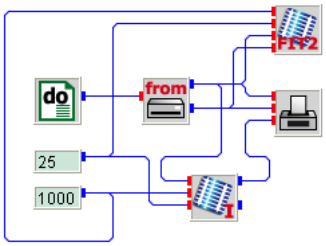
The PVFIT2 block performs a fit to given data points provided by the inputs. The block requires increasing values for the voltage and a unique ($V = 0\,V, I = I_{sc}$) point for each curve. If the short-circuit current is not provided, the result of the PVFIT2 block is unpredictable.

It is recommended to determine the photocurrent parameters c_{ph} and c_t via block PVA2C before using the PVFIT2 block, since the quality of the fit depends very much on these initial values.

Please notice that the block (in its current version) can not be used to determine the two-diode model parameters from operational data but from complete I-V-curves only.

It is not possible to determine the Bishop parameters with the PVFIT2 block.

pvfit2.vseit



A file with measured I-V curve data named `pvfit2.dat`, located in the `examples/data` directory for standard test conditions is read by a READ block driven by a DO block. The radiation and module temperature are set by two CONST blocks to 1000 W/m^2 and 25 °C, respectively.

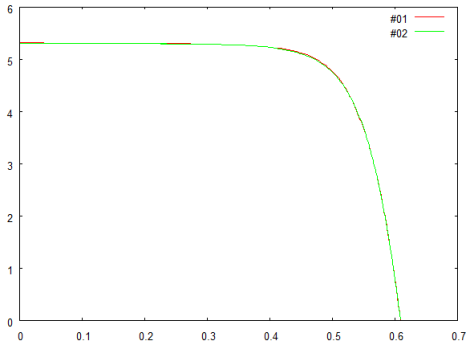
The tupels (G, T, V, I) are fed to the PVFIT2 block which performs the parameter fit. As a result a file named `pvfit2.bp` is generated.

pvfit2.bp

```
                % Filename       pvfit2.bp
                % Module         pvfit2
                % Manufacturer   Manufacturer
                % Cell type      crystalline

                % Mode must be set externally
1               % Number of cells in series N_s per module
1               % Number of cells in parallel N_p per module
1               % Number of modules in series M_s
1               % Number of modules in parallel M_p
0.0156          % Cell area A_c (m^2)
0.016           % Module area A_m (m^2)

                % Electrical parameters
1.12            % Band gap (eV)
0.273035E+00 % Short-circuit current parameter C_0 (V^-1)
0.224599E-03 % Isc temperature coefficient  C_1 (V^-1 K-^1)
0.293476E+04 % Shockley saturation parameter C_01 (A m-2 K^-3)
0.223146E+01 % Recombination saturation parameter C_02 (A m-2 K^-5/2)
0.678692E-04 % Series resistance r_s (Ohm m^2)
0.689963E+00 % Parallel resistance r_p (Ohm m^2)
0.100000E+01 % Shockley diode ideality factor alpha
    2.000000 % Recombination diode quality beta
0.000000E+00 % Bishop parameter-1
0.000000E+00 % Bishop parameter-2
0.000000E+00 % Bishop parameter-3
0.0             % Module tolerance plus
0.0             % Module tolerance minus

                % Thermal parameters
1.000           % Characteristic module length l_m (m)
```

```
  1.000        % Module mass m_m (kg)
  0.70         % Default absorption coefficient a
  0.85         % Default emission factor epsilon
900.0          % Default specific heat of a module C_mod (J kg^-1 K^-1)
 47.0          % Nominal operating cell temperature NOCT (degrees C)
 25.0          % Intial value of cell temperature (degrees C)

               % Numerical parameters (optional)
  1E-5         % Error tolerance of voltage of single cell (V)
100            % Maximal number of iterations to solve I/V-equation
```

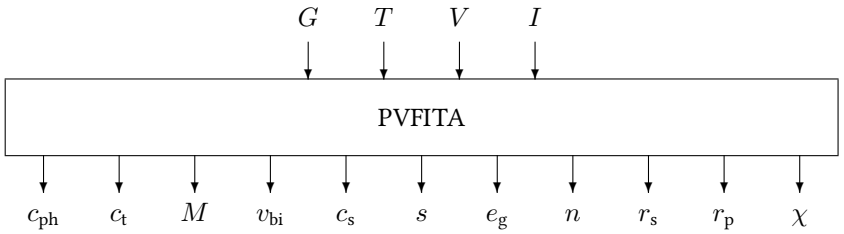
The parameters have been incorporated into the `pvfit2.vseit` example using a PVI block in order to demonstrate the quality of the fit. The PLOT block then shows the measured data (red curve) and the simulated points (green curve).



See also Blocks PVI, PVV, PVFIT1.

3.35 Block PVFITA

The PVFITA block performs a modified one-diode-model fit on I-V curves of a (micro-) amorphous PV module connected to the block's inputs.



Name	PVFITA
Function	se0005
Inputs	4
Outputs	11
Parameters	33 . . . [34]
Strings	5
Group	I

Inputs

1 Global radiation / $\mathrm{W\,m^{-2}}$
2 Module temperature / °C
3 Module voltage / V
4 Module current / A

Outputs

1 Photocurrent parameter c_{ph} / $\mathrm{V^{-1}}$
2 Temperature coefficient c_t / $\mathrm{V^{-1}\,K^{-1}}$
3 Merten parameter M / $\mathrm{V^{-1}}$
4 Built-in voltage parameter v_{bi} / V
5 Saturation current parameter c_s / $\mathrm{A\,m^{-2}\,K^{-s}}$
6 Temperature exponent s
7 Bandgap parameter e_g / eV
8 Diode quality parameter n
9 Series resistance r_s / $\Omega\,\mathrm{m^2}$
10 Parallel resistance r_p / $\Omega\,\mathrm{m^2}$
11 Accuracy χ

Parameters

1 Fit procedure
  0 Nelder Mead downhill simplex method
  1 Marquard Levenberg algorithm (not yet implemented)
2 Number of cells in series
3 Number of cells in parallel
4 Area of a single cell / m^2
5 Module area / m^2
6 Minimum photocurrent parameter / V^{-1}
7 Maximum photocurrent parameter / V^{-1}
8 Minimum temperature coefficient / $V^{-1}K^{-1}$
9 Maximum temperature coefficient / $V^{-1}K^{-1}$
10 Minimum Merten parameter / V^{-1}
11 Maximum Merten parameter / V^{-1}
12 Minimum built-in voltage / V
13 Maximum built-in voltage / V
14 Minimum saturation-current parameter $A\,m^{-2}\,K^{-s}$
15 Maximum saturation-current parameter $A\,m^{-2}\,K^{-s}$
16 Minimum temperature exponent
17 Maximum temperature exponent
18 Minimum bandgap parameter / eV
19 Maximum bandgap parameter / eV
20 Minimum diode-quality parameter
21 Maximum diode-quality parameter
22 Minimum series resistance / $\Omega\,m^2$
23 Maximum series resistance / $\Omega\,m^2$
24 Minimum parallel resistance / $\Omega\,m^2$
25 Maximum parallel resistance / $\Omega\,m^2$
26 Module tolerance plus / $\% > 0$
27 Module tolerance minus / $\% < 0$
28 Characteristic module length / m
29 Module weight / kg
30 Absorption coefficient a
31 Emission coefficient ϵ
32 Specific module heat capacity / $J\,kg^{-1}\,K^{-1}$
33 NOCT temperature / °C
34 Overwrite mode
  0 Generate error (default)
  1 Overwrite

Strings

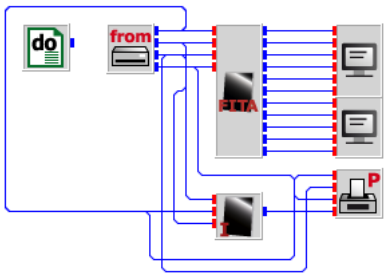
1 INSEL module ID (max. eight characters)
2 Path to .bp file

3 Module name
4 Manufacturer
5 Cell type (e. g., micromorphous)

Description

The PVFITA block can be used to determine the parameters for block PVAI and PVAV, respectively.

pvfita.vseit



A file with measured I-V curve data named `pvfita.dat`, located in the `examples/data` directory for standard test conditions is read by a READ block driven by a DO block. The radiation and module temperature are set by two CONST blocks to 1000 W/m^2 and 25 °C, respectively.

The tupels (G, T, V, I) are fed to the PVFITA block which performs the parameter fit.

Two SCREEN blocks are used to show the parameters.

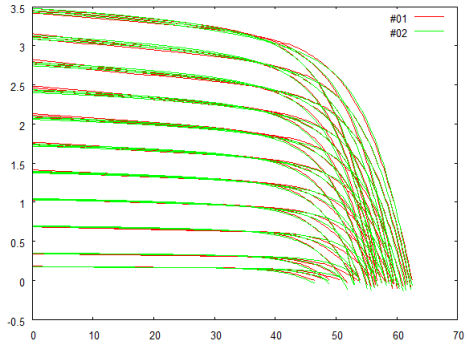
```
0.10807E+00 0.65624E-04 0.58518E+01 0.18123E+01 0.99146E+01 0.23929E+01
0.68577E+00 0.34136E+01 0.65709E-03 0.13827E+02 0.79720E-02
```

As final result a file named `pvfita.bp` is generated.

pvfita.bp

```
                % Filename       pvfita.bp
                % Module         No name
                % Manufacturer   No name
                % Cell type      Mircroamorphous

                % Mode must be set externally
 45             % Number of cells in series N_s per module
  4             % Number of cells in parallel N_p per module
  1             % Number of modules in series M_s
  1             % Number of modules in parallel M_p
  0.0075        % Cell area A_c (m^2)
  1.420         % Module area A_m (m^2)

                % Electrical parameters
  0.108074E+00 % Short-circuit current parameter c_ph (V^-1)
  0.656252E-04 % Isc temperature coefficient c_t (V^-1 K-^1)
  0.585197E+01 % Merten parameter M (V^-1)
  0.181235E+01 % Built-in voltage parameter v_bi (V)
  0.991458E+01 % Coeff. of sat. current density c_s (A m-2 K^-s)
  0.239290E+01 % Temperature exponent of sat. current density s
  0.685773E+00 % Band gap parameter (eV)
      3.413680 % Diode ideality factor
  0.657086E-03 % Series resistance r_s (Ohm m^2)
  0.138270E+02 % Parallel resistance r_p (Ohm m^2)
  5.0           % Module tolerance plus
 -5.0           % Module tolerance minus

                % Thermal parameters
  1.409         % Characteristic module length l_m (m)
 19.000         % Module weight m_m (kg)
  0.70          % Default absorption coefficient a
  0.85          % Default emission factor epsilon
900.0           % Default specific heat of a module C_mod (J kg^-1 K^-1)
 47.0           % Nominal operating cell temperature NOCT (degrees C)
 25.0           % Intial value of cell temperature (degrees C)

                % Numerical parameters
  0.100000E-02 % Error tolerance
100.0           % Maximum number of iterations
```

The parameters have been incorporated into the `pvfita.vseit` example using a PVAI block in order to demonstrate the quality of the fit. The PLOTP block then shows the measured data (red curves) and the simulated points (green curves).



Remark In the current version 8.2 no mechanism is included to create a VSEit entity for a PVAI block with the fitted parameters. A workaround uses the following idea:

Open the generated `.bp` file with a text editor, add these line

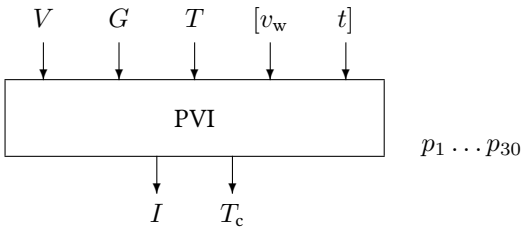
```
s 1 pvai
p 1 0           % Mode
```

to the top of the file, save the file as `somename.ins` and use the File – Import function of VSEit. This will create the wanted PVAI block.

See also Blocks PVAI and PVAV.

3.36 Block PVI

The PVI block calculates the output current and the temperature of a crystalline photovoltaic generator, depending on PV generator voltage, global radiation on the generator plane, ambient temperature, and wind speed.



Name	PVI
Function	se0001
Inputs	3 . . . [5]
Outputs	2
Parameters	30
Strings	0 . . . [1]
Group	S

Inputs

1 Voltage V / V
2 Global radiation on the generator plane G / W m^{-2}
3 Temperature T / °C
4 Wind speed close to the generator v_w / m s^{-1}
5 Time t / s

Outputs

1 Current I / A
2 Cell temperature T_c / °C

Parameters

1 Mode
- 0 The cell temperature is given by input 3.
- 1 The cell temperature depends on the thermal parameters of the PV modules and the inputs of the block; initial value of the cell temperature is given by parameter number 28.
- 2 The difference between cell temperature and ambient temperature is a linear function of the global radiation (input 2) on the basis of NOCT.

2 Number of cells in series per module N_s
3 Number of cells in parallel N_p
4 Number of modules in series M_s
5 Number of modules in parallel M_p
6 Area of a single cell A_c / m^2
7 Area of a single module A_m / m^2
8 Band gap E_g / eV (c-Si $\approx$ 1.12 eV, a-Si $\approx$ 1.75 eV)
9 Coefficient of short-circuit current density c_{ph} / V^{-1}
10 Temperature coefficient of short-circuit current c_t / $V^{-1}\,K^{-1}$
11 Coefficient of saturation current density (Shockley diode) c_s / $A\,m^{-2}\,K^{-3}$
12 Coefficient of saturation current density (Recombination diode) c_r / $A\,m^{-2}\,K^{-5/2}$
13 Series resistance parameter r_s / $\Omega\,m^2$
14 Parallel resistance parameter r_p / $\Omega\,m^2$
15 Diode ideality factor α
16 Diode ideality factor β
17 Bishop parameter a
18 Bishop parameter m
19 Bishop parameter V_{br} / V
20 Module tolerance plus / %
21 Module tolerance minus / % (< 0)
22 Characteristic module length ℓ_m / m
23 Module mass m_m / kg
24 Absorption coefficient a
25 Emission factor ϵ
26 Specific heat of a module c_m / $J\,kg^{-1}K^{-1}$
27 Nominal operating cell temperature NOCT / °C
28 Initial value of cell temperature / °C
29 Error tolerance of voltage (or current) of a single cell in the numerical iteration to solve the two-diode-model equation
30 Maximum number of iterations to solve the two-diode-model equation

Strings

1 Product ID

Description

The model of the PV module has two parts: an electrical model (the “two diode model”) and a thermal model based on an energy balance.

Electrical model

The relationship between voltage V_c / V of a crystalline silicon solar cell and current density j / A m^{-2} is given by the two-diode-model equation

$$\begin{aligned} j = j_{ph} &- j_s \left(\exp\left(\frac{q(V_c + j r_s)}{\alpha k T} \right) - 1 \right) \\ &- j_r \left(\exp\left(\frac{q(V_c + j r_s)}{\beta k T} \right) - 1 \right) - \frac{V_c + j r_s}{r_{sh}} \end{aligned}$$

where

r_s	Series resistance parameter of the cell / Ω m^2
r_{sh}	Shunt resistance parameter / Ω m^2
α	Diode parameter (should be set to 1 in case of the standard two diode model)
β	Diode parameter (should be set to 2 in case of the standard two diode model)
T	Cell temperature / K
q	Charge of an electron (1.6021×10^{-19}As)
k	Boltzmann constant (1.3854×10^{-23}JK^{-1})

Hence, the operating point of the PV generator is given by

$$\begin{aligned} V &= V_c N_s \\ I &= j A_c N_p \end{aligned}$$

where

A_c	Area of a single cell / m^2
N_s	Number of cells in series (whole generator)
N_p	Number of cells in parallel (whole generator)

The light-generated current density j_{ph} / A m^{-2} is proportional to the global radiation G / W m^{-2} on the generator plane and is assumed to be linearly dependent on the cell temperature T / K

$$j_{ph} = (c_{ph} + c_t T)\, G$$

where

c_{ph}	Coefficient of light-generated current density / V^{-1}
c_t	Temperature coefficient of light-generated current density / V^{-1} K^{-1}

The dependence of the saturation current densities j_s and j_r on temperature is given by

$$j_s = c_s T^3 \exp\left(-\frac{qV_{gap}}{kT}\right)$$
$$j_r = c_r T^{5/2} \exp\left(-\frac{qV_{gap}}{2kT}\right)$$

where

c_s	Coefficient of saturation current density / $\mathrm{A\,m^{-2}\,K^{-3}}$
c_r	Coefficient of saturation current density / $\mathrm{A\,m^{-2}\,K^{-5/2}}$
V_{gap}	Band gap (Silicon 1.12 V). The dependence of the band gap on cell temperature is neglected.

Thermal model

The thermal model is based on an energy balance

$$m_{mod} N_{mod} c_{mod} \frac{dT}{dt} + P_{el} = \dot{Q}_G - \dot{Q}_r - \dot{Q}_c$$

where

m_{mod}	Mass of a PV module / kg
c_{mod}	Specific heat of a PV module $\mathrm{J\,kg^{-1}\,K^{-1}}$
P_{el}	Electrical power output of the generator / W
$\dot{Q}_G$	Insolation on the whole generator / W
$\dot{Q}_r$	Losses through radiation / W
$\dot{Q}_c$	Losses through convection / W

The absorbed insolation is modeled by

$$\dot{Q}_G = aG(t)A_{mod}N_{mod}$$

The coefficient a of absorption is assumed to be constant. Losses through radiation are modeled by

$$\dot{Q}_r = 2\epsilon A_{mod} N_{mod} \sigma (T^4 - {T_a}^4)$$

where

ϵ	Emission factor
σ	Stefan-Boltzmann Constant ($5.6697 \times 10^{-8} \mathrm{Wm^{-2}K^{-4}}$)
T_a	Ambient temperature / K

Losses through convection are given by

$$\dot{Q}_c = 2\gamma A_{mod} N_{mod} (T - T_a)$$

In case of free convection the heat loss coefficient γ is set to

$$\gamma_f = 1.78\,(T - T_a)^{1/3}$$

forced convection is modeled through

$$\gamma_w = \frac{4.77\,{v_w}^{0.8} (\ell_{mod} N_{mod})^{-0.2}}{1 - 0.17\,{v_w}^{-0.1} (\ell_{mod} N_{mod})^{-0.1}}$$

with wind speed v_w. In case of mixed convection γ is set to

$$\gamma = \sqrt[3]{{\gamma_f}^3 + {\gamma_w}^3}$$

NOCT mode

According to U.S. standards the nominal operating cell temperature is defined as the temperature of a PV module operated in its maximum power point under 800 W/m^2 irradiance at an ambient temperature of 20 °C and a wind speed of 1 m/s.

The overtemperature – i. e., the temperature difference between module temperature T_m and ambient temperatur T_a – of a PV module in NOCT mode is calculated from the equation

$$\Delta T = T_m - T_a = (T_{NOCT} - 20°C) * G_t / 800\,W/m^2$$

where T_{NOCT} denotes the nominal operating cell temperature, and G_t the irradiance in the module plane.

Please notice that the small temperature dependency as a function of ambient temperature (of about 5 % per 100 °C) is neglected in this mode.

Backward characteristic

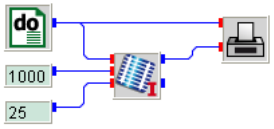
The PVI block can simulate the backward characteristics of a module, if the Bishop parameters a, V_{br}, and m are known. In this case the two-diode-model equation is expanded by the Bishop term

$$\frac{V_c + j r_s}{r_{sh}} \left(1 + a \left(1 - \frac{V_c}{V_{br}}\right)^{-m}\right)$$

If all three Bishop parameters are set to zero, the PVI block suppresses negative currents, i. e., if $I < 0$ then I is set to zero.

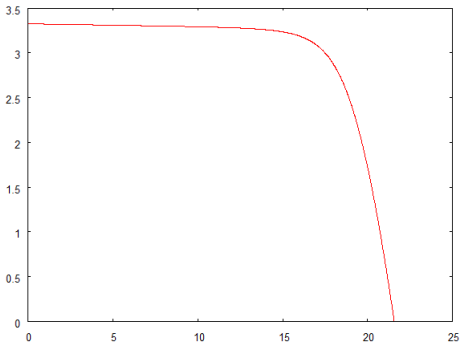
Remarks It is possible to use the one-diode model by setting $c_r = 0$. Parameters for about 5000 modules are given in standard parameter .bp files and can be used in the PVI block's entity editor.

pvi.vseit



A DO block is used to vary the voltage of a PV module from 0 to 25 V in steps of 0.01 V. To simulate standard test conditions two CONST blocks are used to set the module temperature to 25 °C and the radiation to 1000 W/m^2, respectively.

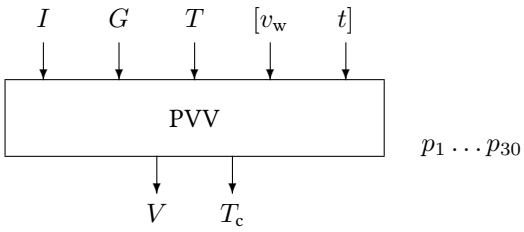
The PLOT block displays the I-V curve for STC.



See also Blocks PVV, PVDET1, PVDET2, PVFIT1, PVFIT2.

3.37 Block PVV

The PVV block calculates the output voltage and the temperature of a crystalline photovoltaic generator, depending on PV generator current, global radiation on the generator plane, ambient temperature, and wind speed.



Name	PVV
Function	se0001
Inputs	3 . . . [5]
Outputs	2
Parameters	30
Strings	0 . . . [1]
Group	S

Inputs

1 Current I / A
2 Global radiation on the generator plane G / W m^{-2}
3 Temperature T / °C
4 Wind speed close to the generator v_w / m s^{-1}
5 Time t / s

Outputs

1 Voltage V / V
2 Cell temperature T_c / °C

Parameters

1 Mode
- 0 The cell temperature is given by input 3.
- 1 The cell temperature depends on the thermal parameters of the PV modules and the inputs of the block; initial value of the cell temperature is given by parameter number 28.
- 2 The difference between cell temperature and ambient temperature is a linear function of the global radiation (input 2) on the basis of NOCT.

2 Number of cells in series per module N_s
3 Number of cells in parallel N_p
4 Number of modules in series M_s
5 Number of modules in parallel M_p
6 Area of a single cell A_c / m^2
7 Area of a single module A_m / m^2
8 Band gap E_g / eV (c-Si $\approx$ 1.12 eV, a-Si $\approx$ 1.75 eV)
9 Coefficient of short-circuit current density c_{ph} / V^{-1}
10 Temperature coefficient of short-circuit current c_t / $V^{-1}\,K^{-1}$
11 Coefficient of saturation current density (Shockley diode) c_s / $A\,m^{-2}\,K^{-3}$
12 Coefficient of saturation current density (Recombination diode) c_r / $A\,m^{-2}\,K^{-5/2}$
13 Series resistance parameter r_s / $\Omega\,m^2$
14 Parallel resistance parameter r_p / $\Omega\,m^2$
15 Diode ideality factor α
16 Diode ideality factor β
17 Bishop parameter a
18 Bishop parameter m
19 Bishop parameter V_{br} / V
20 Module tolerance plus / %
21 Module tolerance minus / % (< 0)
22 Characteristic module length ℓ_m / m
23 Module mass m_m / kg
24 Absorption coefficient a
25 Emission factor ϵ
26 Specific heat of a module c_m / $J\,kg^{-1}K^{-1}$
27 Nominal operating cell temperature NOCT / °C
28 Initial value of cell temperature / °C
29 Error tolerance of voltage (or current) of a single cell in the numerical iteration to solve the two-diode-model equation
30 Maximum number of iterations to solve the two-diode-model equation

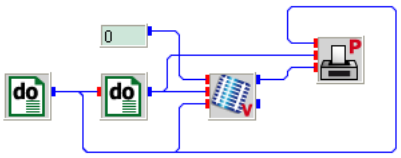
Strings

1 Product ID

Description

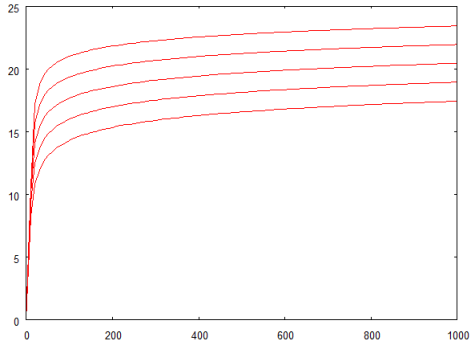
For a detailed description of the models used in the PVV block please refer to the description of block PVI.

pvv.vseit



Two nested DO blocks are used to vary temperature (outer block) and radiation (inner block) from 0 to 80 °C in steps of 20 °C and 0 to 1000 W/m^2 in steps of 10 W/m^2, respectively. The current input of the PVV block is set to zero by a CONST block in order to simulate open-circuit voltage.

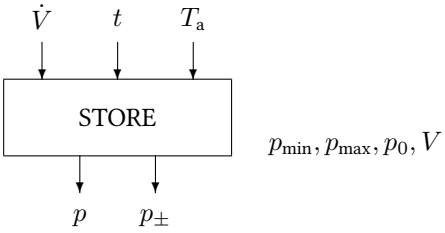
The PLOTP block displays radiation dependence of the module's open-circuit voltage with the module temperature as curve parameter.



See also Blocks PVI, PVDET1, PVDET2, PVFIT1, PVFIT2.

3.38 Block STORE

The STORE block simulates a pressure storage with a safety valve.



Name	STORE
Function	se0015
Inputs	3
Outputs	2
Parameters	4
Strings	0
Group	S

Inputs

1 Volume flow $\dot{V}$ / $m_n^3\,s^{-1}$
2 Time t / s
3 Ambient temperature T_a / °C

Outputs

1 Tank pressure p / bar
2 Difference pressure $p_\pm$ / bar

Parameters

1 Minimum tank pressure p_{min} / bar
2 Maximum tank pressure p_{max} / bar
3 Initial tank pressure p_0 / bar
4 Tank volume V / m^3

Strings

None

Description

It is assumed that the gas is an ideal gas and the tank volume V is a constant. Hence, the volume flow $\dot{V}$ for a time interval Δt, which is calculated from the time input, causes a change in the actual tank pressure

$$p_{\text{new}} = p + \frac{(T_{\text{a}} + T_{\text{n}})\, p_{\text{n}}}{T_{\text{n}}\, V}\, \dot{V} \Delta t \tag{3.1}$$

where

T_{n} Standard temperature 273.15 K

p_{n} Standard pressure 1.013 bar

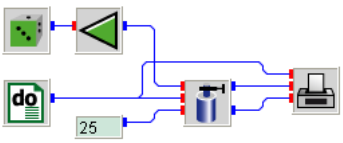
The outputs are set to

$$p = \begin{cases} p_{\text{min}} & \text{if } p_{\text{new}} < p_{\text{min}} \\ p_{\text{new}} & \text{if } p_{\text{min}} \le p_{\text{new}} \le p_{\text{max}} \\ p_{\text{max}} & \text{if } p_{\text{new}} > p_{\text{max}} \end{cases}$$

and

$$p_{\pm} = \begin{cases} p_{\text{new}} - p_{\text{min}} & \text{if } p_{\text{new}} < p_{\text{min}} \\ 0 & \text{if } p_{\text{min}} \le p_{\text{new}} \le p_{\text{max}} \\ p_{\text{new}} - p_{\text{max}} & \text{if } p_{\text{new}} > p_{\text{max}} \end{cases}$$

store.vseit

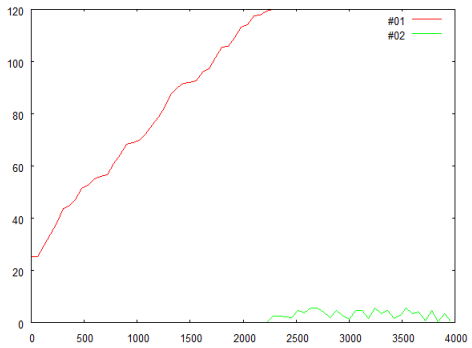


A RAN1 block generates uniform random numbers diluted by a factor of 10 by an ATT block, finally fed into a STORE block as volume flow in m^3/s. The storage tank is set to an initial pressure of 25 bar, with a tank volume of 1.2 m^3.

A DO block runs the simulation for 4000 seconds with a time step of 60 seconds. The time signal is connected to the STORE block.

The temperature of the high pressure storage is set to 25 °C by a CONST block.

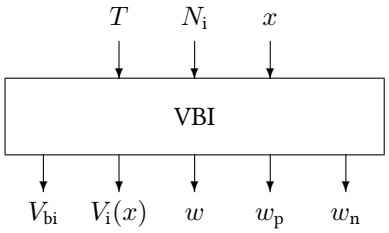
The PLOT block displays tank pressure and the difference pressure which has to be dumped via the safety valve.



See also Block BALON.

3.39 Block VBI

The VBI block calculates the built-in voltage of a pn-junction.



Name	VBI
Function	se0032
Inputs	2 . . . [3]
Outputs	5
Parameters	3
Strings	0
Group	S

Inputs

1	Temperature / °C
2	Intrinsic carrier density / cm^{-3}
3	Position x in pn-junction / μm

Outputs

1	Built-in voltage / V
2	Built-in voltage at x / V
3	Width of the depletion region / μm
4	Width of the depletion region in p-type material / μm
5	Width of the depletion region in n-type material / μm

Parameters

1	Number of acceptor atoms N_a / cm^{-3}
2	Number of donator atoms N_d / cm^{-3}
3	Relative permittivity ϵ_r of the semiconductor (e. g., c-Si 11.68)

Strings

None

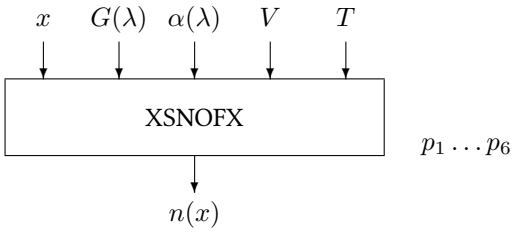
vbi.vseit

The built-in voltage for 10^{17} acceptor and 10^{17} donator atoms in a c-Si crystal is about 0.8 volt as displayed by the SCREEN block.

```
0.82823104
```

3.40 Block XSNOFX

The XSNOFX block calculates the excess electrons in p-type silicon as a function of the material depth.



Name	XSNOFX
Function	se0023
Inputs	5
Outputs	4
Parameters	6
Strings	0
Group	S

Inputs

1 Depth x / cm
2 Generation rate $G(\lambda)$ / $\mathrm{cm^{-3}\ s^{-1}}$
3 Absorption coefficient $\alpha(\lambda)$
4 Voltage V / V
5 Temperature T / °C

Outputs

1 Number of excess electrons $n(x)$

Parameters

1 Equilibrium electron concentration in p-type Si n_{po} / $\mathrm{cm^{-3}}$
2 Diffusion constant of electrons D_{n} / $\mathrm{cm^2\ s^{-1}}$
3 Diffusion length of electrons L_{n} / μm
4 Thickness of basis d_{ba} / cm
5 Thickness of emitter d_{em} / cm
6 Recombination velocity at backside s_{n} / $\mathrm{cm\ s^{-1}}$

Strings

None

Remarks

Solution from Wagemann/Eschrich Appendix A2, Eqn. A2.11)

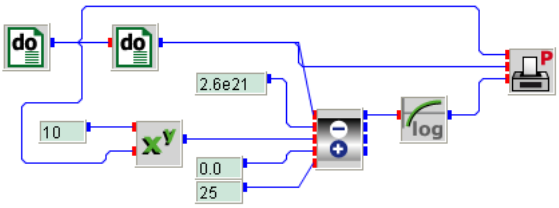
Description

According to Wagemann the number of excess electrons in the basis of a crystalline silicon semiconductor can be calculated after

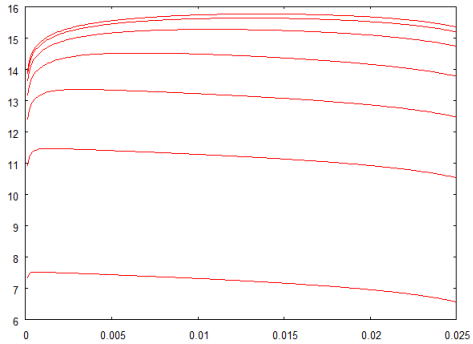
$$\Delta n(x) = n_{\mathrm{po}} \left(\exp\left(\frac{qV}{kT}\right) - 1 \right) \cdot \frac{\frac{D_\mathrm{n}}{L_\mathrm{n}} \cosh(x') + s_\mathrm{n} \sinh(x')}{\frac{D_\mathrm{n}}{L_\mathrm{n}} \cosh(x_0) + s_\mathrm{n} \sinh(x_0)}$$
$$+ \frac{G_0(\lambda)\tau_\mathrm{n}}{1-\alpha(\lambda)^2 L_\mathrm{n}^2} \exp(-\alpha(\lambda) d_\mathrm{em}) \cdot \Bigg(\exp(-\alpha(\lambda)x)$$
$$+ \frac{(D_\mathrm{n}\alpha(\lambda) - s_\mathrm{n}) \exp(-\alpha(\lambda) d_\mathrm{ba} \sinh(\frac{x}{L_\mathrm{n}}) - \frac{D_\mathrm{n}}{L_\mathrm{n}} \cosh(x') - s_\mathrm{n} \sinh(x')}{\frac{D_\mathrm{n}}{L_\mathrm{n}} \cosh(x_0) + s_n \sinh(x_0)} \Bigg)$$

with the shortcuts $x' = \frac{d_\mathrm{ba}-x}{L_\mathrm{n}}$ and $x_0 = \frac{d_\mathrm{ba}}{L_\mathrm{n}}$.

xsnofx.vseit



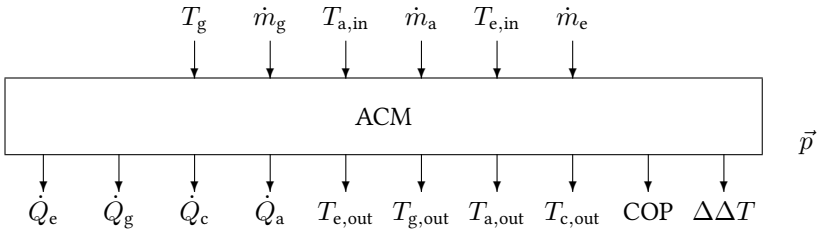
For typical values of a doped silicon crystal the PLOT block shows the logarithmic number of excess electrons as a function of the depth in the basis.



4 :: Solar Thermal Energy

4.1 Block ACM

The ACM block simulates the absorption cycle process of a single- stage, indirectly heated absorption cooling machine with the refrigerant water and the solvent lithium bromide.



Name	ACM
Function	st0026
Inputs	6
Outputs	10
Parameters	15
Strings	0
Group	3

Inputs

1 External generator inlet temperature $T_{g,in}$ / °C
2 External massflow heating circuit $\dot{m}_{g,in}$ / kg s^{-1}
3 External absorber inlet temperature $T_{a,in}$ / °C
4 External massflow cooling circuit at absorber/condenser $\dot{m}_{a,in}$ / kg s^{-1}
5 External evaporator inlet temperature $T_{e,in}$ / °C
6 External massflow cooling circuit $\dot{m}_{e,in}$ / kg s^{-1}

Outputs

1 Refrigerating capacity of the evaporator $\dot{Q}_e$ / kW
2 Heating capacity of the generator $\dot{Q}_g$ / kW
3 Cooling capacity of the condenser $\dot{Q}_c$ / kW
4 Cooling capacity of the absorber $\dot{Q}_a$ / kW
5 Outlet temperature cooling circuit $T_{e,out}$ / °C
6 Outlet temperature heating circuit $T_{g,out}$ / °C
7 Outlet temperature cooling circuit absorber $T_{a,out}$ / °C
8 Outlet temperature cooling circuit condenser $T_{c,out}$ / °C
9 Coefficient of performance COP

10 Characteristic double temperature difference $\Delta\Delta T$ / K

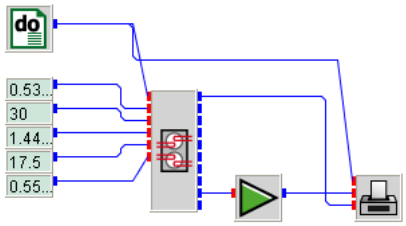
Parameters

1 Initial internal generator mean temperature $T_{g,0}$ / °C
2 Initial internal condenser mean temperature $T_{c,0}$ / °C
3 Initial internal absorber mean temperature $T_{a,0}$ / °C
4 Initial internal evaporator mean temperature $T_{e,0}$ / °C
5 Characteristic number for the generator heat transfer $(UA)_g$ / kW K^{-1}
6 Characteristic number for the condenser heat transfer $(UA)_c$ / kW K^{-1}
7 Characteristic number for the absorber heat transfer $(UA)_a$ / kW K^{-1}
8 Characteristic number for the evaporator heat transfer $(UA)_e$ / kW K^{-1}
9 Rich solution mass flow $\dot{m}_r$ / kg s^{-1}
10 Solution heat exchanger efficiency η_s
11 Heat capacity external heating fluid $c_{p,g}$ / kJ kg^{-1} K^{-1}
12 Heat capacity external cooling fluid $c_{p,a}$ / kJ kg^{-1} K^{-1}
13 Heat capacity external cooling circle fluid $c_{p,e}$ / kJ kg^{-1} K^{-1}
14 Dühring-Parameter B
15 Cooling capacity under design conditions $\dot{Q}_n$ / kW

Strings

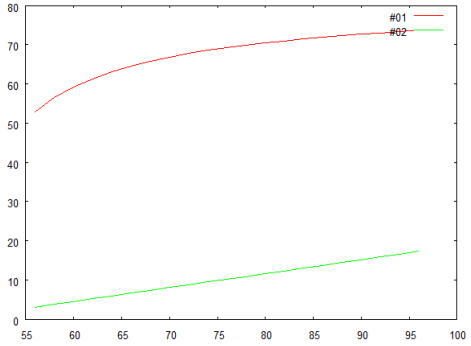
None

acm.vseit



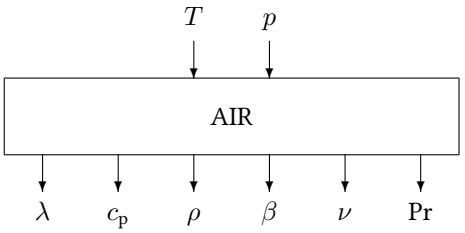
A DO blocks varies the external generator inlet temperature of an absorption cooling machine between 56 and 96 °C in steps of 2 °C. All other inputs of the ACM block are set constant through five CONST blocks. The coefficient of performance of the ACM is multiplied by a factor 100, i. e., converted into percent by a GAIN block.

The COP together with the refrigeration capacity of the ACM is displayed by a PLOT block.



4.2 Block AIR

The AIR block handles properties of air as a function of temperature and pressure.



Name	AIR
Function	st0012
Inputs	2
Outputs	6
Parameters	0
Strings	0
Group	S

Inputs

1 Temperature T / °C
2 Pressure p / Pa

Outputs

1 Heat conductivity λ / $\mathrm{W\,m^{-1}\,K^{-1}}$
2 Specific heat capacity c_p / $\mathrm{J\,kg^{-1}\,K^{-1}}$
3 Density ρ / $\mathrm{kg\,m^{-3}}$
4 Heat expansion coefficient β / $\mathrm{K^{-1}}$
5 Viscosity ν / $\mathrm{m^2\,s^{-1}}$
6 Prandtl number Pr

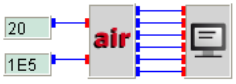
Parameters

None

Strings

None

air.vseit

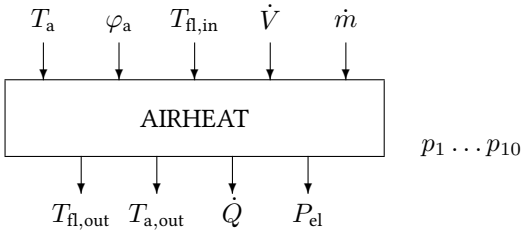


Two CONST blocks provide a temperature of 20 °C and a pressure of 10^5 Pa. The AIR block calculates the properties which are displayed by a SCREEN block.

```
0.02512    1007.00000       1.18817       0.00341   0.15175E-04       0.72288
```

4.3 Block AIRHEAT

The AIRHEAT block simulates heating performance of a hot water to air heater as well as the cooling performance of a dry cooling tower in dependency of the perfromance data given on the data sheet for nominal operation conditions. Only valid for water volume flow rates of at least 30 percent of the nominal water flow rate.



Name	AIRHEAT
Function	st0037
Inputs	5
Outputs	4
Parameters	10
Strings	0
Group	S

Inputs

1 Ambient air temperature / °C
2 Relative humidity air
3 Inlet water temperature / °C
4 Air flow rate in percent of nominal air flow rate (20 – 100 %)
5 Water mass flow rate / kg s^{-1}

Outputs

1 Water outlet temperature / °C
2 Air outlet temperature / °C
3 Recooling / heating power of the cooling tower / kW
4 Required electricity power of the ventilators / kW

Parameters

1 Nominal recooling/heating power of the air heater / dry cooling tower / kW
2 Heating fluid inlet temperature under design conditions / °C
3 Heating fluid mass flow rate under design conditions / kg s^{-1}

4	Density of heating fluid / kg m^{-3}
5	Specific heat of heating fluid / kJ kg^{-1} K^{-1}
6	(Ambient) air inlet temperature under design conditions / °C
7	Air volume flow under design conditions / m^3 h^{-1}
8	Electrical power consumption of the ventilator motor under design conditiones / kW
9	Total length of heat exchanger unit / m
10	Total width of heat exchanger unit / m

Strings

None

Description

The AIRHEAT block simulates the heating performance of a hot water to air heater as well as the cooling performance of a dry cooling tower in dependency of the performance data given on the data sheet for nominal operation conditions. In this model the outlet temperature of the air flow as well as the electrical power consumption of the fan are calculated in the dependency of the inlet parameters and the given nominal heating/cooling power. Because of the assumption made within the modeling, the model is only valid for water volume flow rates of at least 30 percent of the nominal water flow rate.

Fundamentals

The overall energy balance for the heating block is given by the following equation

$$\dot{m}_{\mathrm{fl}} \cdot c_{\mathrm{p,fl}} \cdot (T_{\mathrm{fl,in}} - T_{\mathrm{fl,out}}) = \dot{m}_{\mathrm{a}} \cdot c_{\mathrm{p,a}} \cdot (T_{\mathrm{a,out}} - T_{\mathrm{a,in}})$$

where

T_{in}	Air (a) and fluid (fl) inlet temperature / °C
T_{out}	Air (a) and fluid (fl) outlet temperature / °C
c_{p}	Specific heat capacity of air (a) and fluid (fl) / kJ kg^{-1} K^{-1}
$\dot{m}$	Mass flows of air (a) and fluid (fl) / kg s^{-1}

Assuming constant mass flow rates the heat exchanger efficiency is defined as follows

$$\phi = \frac{T_{\mathrm{a,out}} - T_{\mathrm{a,in}}}{T_{\mathrm{fl,in}} - T_{\mathrm{a,in}}}$$

If the specific heat capacity of the fluid is much higher then the one of the air, a middle temperature can be assumed resulting in the following equation

$$\phi = \frac{T_{\mathrm{a,out}} - T_{\mathrm{a,in}}}{T_{\mathrm{H}} - T_{\mathrm{a,in}}}$$

$$T_{\mathrm{H}} = \frac{T_{\mathrm{fl,in}} + T_{\mathrm{fl,out}}}{2}$$

For constant average fluid temperature, due to a larger heat capacity, the heat exchanger efficiency is also given by

$$\phi = 1 - \exp\left(\frac{U \cdot A}{\dot{m}_\mathrm{a} \cdot c_\mathrm{p,a}}\right)$$

where

U	Heat transfer coefficient of the heat exchanger / $\mathrm{W\,m^{-2}\,K^{-1}}$
A	Heat transfer area inside the heat exchanger / $\mathrm{m^2}$

The UA-value of the heat exchanger can be calculated at design conditions. For changing massflow rates the following law can be used

$$U \cdot A = C_\mathrm{HX} \cdot v_\mathrm{a}^n$$

with

v	Air velocity / $\mathrm{m\,s^{-1}}$
n	Parameter $n = 0.4 \ldots 0.6$
C_HX	Characteristic heat exchanger constant / $\mathrm{J\,m^{-3}\,K^{-1}}$

In order to calculate the air properties, empirical equations are used

$$\begin{aligned} p_\mathrm{s,a} = {} & 611 \cdot \exp(-1.91275 \cdot 10^{-4} + T_\mathrm{a} \cdot 7.258 \cdot 10^{-2} \\ & - T_\mathrm{a}^2 \cdot 2.939 \cdot 10^{-4} + T_\mathrm{a}^3 \cdot 9.843 \cdot 10^{-7} \\ & - T_\mathrm{a}^4 \cdot 1.92 \cdot 10^{-9}) \end{aligned}$$

$$\rho_\mathrm{a} = \frac{1}{T_\mathrm{a} + 273.15} \cdot \left(p - \frac{\varphi_\mathrm{a} \cdot p_\mathrm{s,a}}{R_\mathrm{a}} + \frac{\varphi_\mathrm{a} \cdot p_\mathrm{s,a}}{R_\mathrm{d}}\right)$$

with

$p_\mathrm{s,a}$	Air saturation pressure / Pa
ρ_a	Air density / $\mathrm{kg\,m^{-3}}$
R_a	Universal gas constant air ($287\ \mathrm{J\,kg^{-1}\,K^{-1}}$)
R_d	Universal gas constant steam ($462\ \mathrm{J\,kg^{-1}\,K^{-1}}$)

Before describing the algorithm which makes it possible to calculate the outlet temperatures, the proportional law for determining the electric power consumption of the ventilator is given.

$$P_\mathrm{el} = P_\mathrm{el,d} \cdot \left(\frac{\dot{V}}{\dot{V}_\mathrm{max}}\right)$$

with

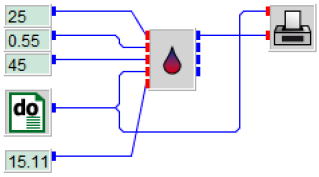
P_{el}	Electric power consumption / W
$P_{\text{el,d}}$	Electric power consumption at design conditions / W
$\dot{V}$	Air volume flow / $\text{m}^3\,\text{s}^{-1}$

Implementation

This subsection presents the algorithm used in order to describe the heat/ cooling block. In contrast to other heat exchanger models this block calculates the output via a given heat exchange capacity at nominal conditions.

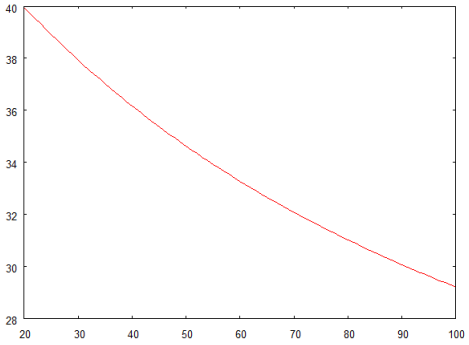
1. Calculate air properties at design conditions
2. Determine air and water output temperatures with the input values as well as the air speed at design conditions in order to calculate the heat exchanger characteristic $U \cdot A$ and for C_{HX}
3. Calculate air properties at working conditions
4. Correct air flow at design $\dot{V}_{\text{a}} = n_{\text{rel}} \cdot \dot{V}_{\text{a,d}}$ conditions via the relative revolutions per second with $n_{\text{rel}} = \frac{n}{n_{\text{max}}}$ and calculate the air velocity at working conditions
5. Calculate the heat transfer $U \cdot A$ at working conditions
6. With $U \cdot A$ calculate ϕ
7. Now using the outlet temperatures of the air and the fluid can be calculated
8. Calculate the electric power consumption using the proportional law $\frac{\dot{V}}{\dot{V}_{\text{max}}} = \frac{n}{n_{\text{max}}}$

airheat.vseit



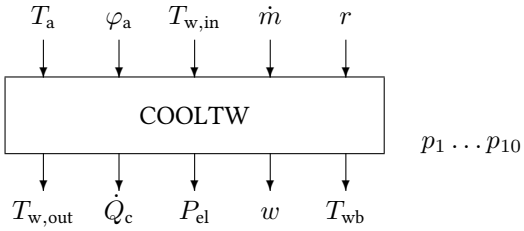
A DO block varies the air flow rate of the heater between 20 and 100 percent of the nominal airflow. All other inputs are set constant.

A PLOT block shows the water outlet temperature as a function of the air flow rate.



4.4 Block COOLTW

The COOLTW simulates the cooling performance, the electricity and water consumption of an open cooling tower in counter-flow arrangement in dependency of the fill packing size and the perfromance data given on the data sheet for nominal operation conditions



Name	COOLTW
Function	st0030
Inputs	5
Outputs	5
Parameters	10
Strings	0
Group	S

Inputs

1 Ambient air temperature / °C
2 Relative humidity ambient air
3 Inlet water temperature / °C
4 Water mass flow rate / kg s^{-1}
5 Air mass flow rate in percent of nominal mass flow rate (20 - 100 %) / %

Outputs

1 Outlet water temperature / °C
2 Recooling power of the cooling tower / kW
3 Required electricity power of the ventilator / kW
4 Overall water consumption / kg h^{-1}
5 Wet bulb temperature ambient air / °C

Parameters

1 Nominal recooling power of the cooling tower / kW
2 Water inlet temperature under design conditions / °C
3 Water outlet temperature under design conditions / °C

4	Water mass flow rate under design conditions / kg s^{-1}
5	Ambient air temperature under design conditions / °C
6	Wet bulb temperature under design conditions / °C
7	Electrical power consumption of the ventilator motor under design conditions / kW
8	Height of fill packing / m
9	Water consumption through spray losses per kW cooling power / kg h^{-1}
10	Water exchange rate per kW cooling power to avoid inrease of salt concentration / kg h^{-1}

Strings

None

Description

The COOLTW simulates the cooling performance, the electricity and water consumption of an open cooling tower in counter-flow arrangement in dependency of the fill packing size and the performance data given on the data sheet for nominal operation conditions.

In wet cooling towers mostly latent heat transfer is used to lower the temperature of the cooling fluid. Open cooling towers generally work with water as cooling fluid which is distributed over a fill packing and forced air-cooled by a ventilator. Partial evaporation of approximately 2–3% of the water as well as convective heat transfer between water and ambient air result in the cooling effect. Due to the evaporation of water outlet temperatures of the cooling water blow the ambient air temperature can be reached, which are only limited by the wet bulb temperature of the ambient air.

Fundamentals

The effect of cooling water by evaporation, like it is used in open cooling towers, will at first be described for an ideal process. Therein an air mass flow is passed over a free surface of water having the same temperature as the boundary water layer. Because of atomic interactions a water mass flow sets in motion and cools down the water. Thus overall energy balance can be described as follows.

$$\dot{m}_{\mathrm{w}} \cdot c_{\mathrm{p,w}} \cdot (T_{\mathrm{w,in}} - T_{\mathrm{w,out}}) = \dot{m}_{\mathrm{a}} \cdot (h_{\mathrm{a,out}} - h_{\mathrm{a,in}})$$

where

$T_{\mathrm{w,in}}$	Water inlet temperature / °C
$T_{\mathrm{w,out}}$	Water outlet temperature / °C
$h_{\mathrm{a,in}}$	Air inlet enthalpy / kJ kg^{-1}
$h_{\mathrm{a,out}}$	Air outlet enthalpy / kJ kg^{-1}
$c_{\mathrm{p,w}}$	Water heat capacity / kJ kg^{-1} K^{-1}
$\dot{m}$	Mass flows of air (a) and water (w) / kg s^{-1}

Modeling

For an ideal process the temperature that can be reached, until the convective heat flow from air to water compensates for the latent heat flow from water to air via mass flow interaction, is called the air's wet bulb temperature t_{wb}. Because this temperature can't be reached for a finite surface and time span, the recooling efficiency is introduced.

$$\eta = \frac{T_{\mathrm{w,in}} - T_{\mathrm{w,out}}}{T_{\mathrm{w,in}} - T_{\mathrm{wb}}}$$

with

T_{wb}	Wet bulb temperature / °C

Another parameter of wet cooling towers is the so called air ratio λ, which defines the relation between the used and the minimal air mass flow. It can be calculated from the effective and minimum air to water mass flow ratios l_0 and l_{min}.

$$l_{\mathrm{min}} = \frac{\dot{m}_{\mathrm{a,min}}}{\dot{m}_{\mathrm{w}}}$$

$$l_0 = \frac{\dot{m}_{\mathrm{a}}}{\dot{m}_{\mathrm{w}}}$$

$$\lambda = \frac{l_0}{l_{\mathrm{min}}}$$

with

$\dot{m}_{\mathrm{a,min}}$	Minimum air mass flow rate / kg s^{-1}
$\dot{m}_{\mathrm{a}}$	Air mass flow rate / kg s^{-1}
$\dot{m}_{\mathrm{w}}$	Water mass flow rate / kg s^{-1}
l_{min}	Minimum air to water mass flow ratio
l_0	Effective air to water mass flow ratio
λ	Air ratio

Also there is a clear correlation between the recooling efficiency η and the air ratio. The characteristic behavior can be described using the following equation

$$\eta = C_{\mathrm{k}} \cdot (1 - \exp(-\lambda))$$

with

C_{k}	Cooling tower constant

After presenting the cooling tower constant C_{k}, which can be calculated using design conditions, a simple open cooling tower model can be built. The advantage of using the presented equation is the minimal number of necessary parameters. In order to even simplify the parameterization, an empirical relationship between the cooling tower

constant and the height of the fill packing H has been derived from typical values and is now used in this model.

$$C_{\mathrm{k}} = -0.5625 \cdot H^2 + 1.26 \cdot H + 0.3221$$

According to the proportional law for axial and radial ventilators the following general correlation of the electricity power consumption to the third power of the fan speed is given.

$$P_{\mathrm{el}} = P_{\mathrm{el,d}} \cdot \left(\frac{n}{n_{\mathrm{d}}} \right)^3$$

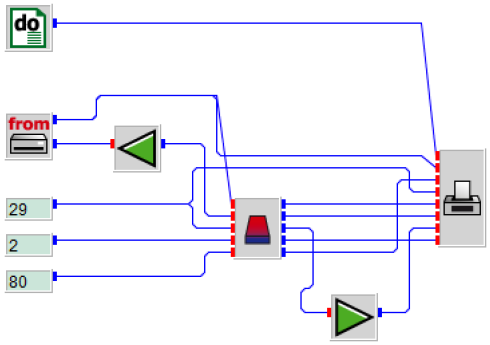
P_{el}	Electric power demand at n / W
$P_{\mathrm{el,d}}$	Electric power demand at design conditions / W
n_{d}	Number of revolutions per second of the cooling tower fan / s^{-1}
n_{d}	Number of revolutions per second of the cooling tower fan at design conditions / s^{-1}

Implementation

This subsection presents the algorithm used in order to calculate the open cooling tower.

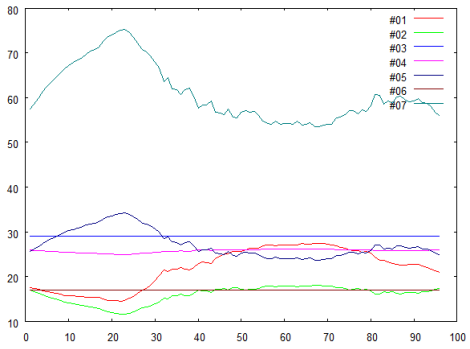
1. Parameters from design (d) conditions: $l_{\mathrm{min,d}} \rightarrow \eta_{\mathrm{d}} \rightarrow \lambda_{\mathrm{d}} \rightarrow l_{\mathrm{0,d}}$
2. Calculate the air mass flow: $\dot{m}_{\mathrm{a}} = l_{\mathrm{min,d}} \cdot \dot{m}_{\mathrm{w,d}} \cdot \frac{n}{n_{\mathrm{max}}}$ with the number of revolutions per second of the fan n
3. Determine the new effective air to water mass flow ratio $l_0 = \frac{\dot{m}_{\mathrm{a}}}{\dot{m}_{\mathrm{w}}}$
4. Estimate the wet bulb temperature with the assumption of isenthalpic humidification via iteration $x_{\mathrm{wb}} = f(\varphi(x), T(x))$ and a relative humidity $\varphi = 100\%$
5. Calculate the working condition parameters with the wet bulb temperature $l_{\mathrm{min}} \rightarrow \eta \rightarrow \lambda \rightarrow T_{\mathrm{w,out}}$
6. Estimate the air outlet temperature with the assumption of isenthalpic humidification via iteration $x_{\mathrm{a,out}} = f(\varphi(x), T(x))$
7. Calculate electric power consumption at n

cooltw.vseit



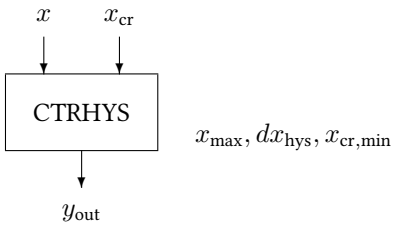
A DO block drives a READ block to read 96 records from the file `iseyear7.dat`, located in the `examples/data` directory. The records contain ambient temperature data in degrees Celsius and relative humidity in percent. An ATT block normalizes the percent values as required by the COOLTW block.

A PLOT block shows all outputs of the COOLTW block as a function of time.



4.5 Block CTRHYS

The CTRHYS block simulates a differential controller that generates a control function and returns the values 0 or 1. A critical value can be controlled for safety reasons.



Name	CTRHYS
Function	st0036
Inputs	2
Outputs	1
Parameters	3
Strings	0
Group	S

Inputs

1 Value to be controled x
2 Critical value (lower limit) x_{cr}

Outputs

1 Output control function y_{out}

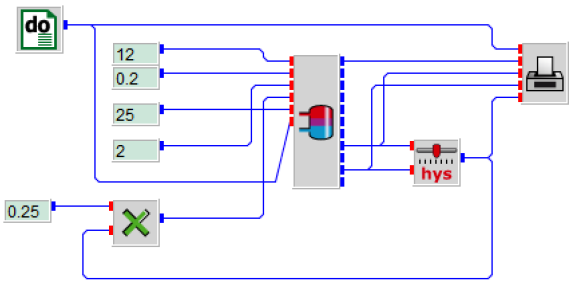
Parameters

1 Max. value to switch on x_{max}
2 Hysteresis to switch off dx_{hys}
3 Critical value min limit $x_{cr,min}$

Strings

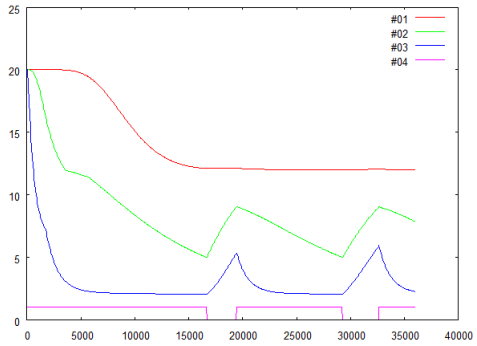
None

ctrhys.vseit



In this example a stratified storage tank is controlled by a CTRHYS block. The time is simulated by a DO block which runs for ten hours in steps of 60 seconds. The operation conditions are set constant.

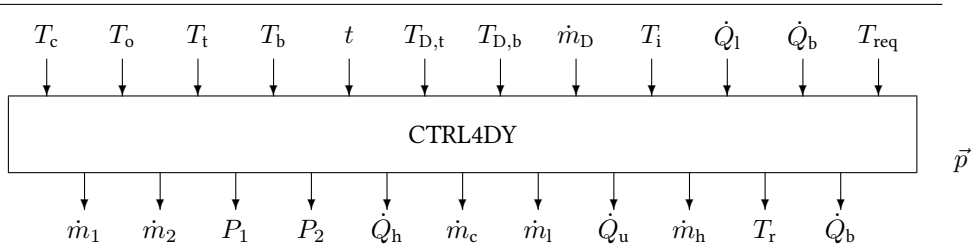
A PLOT block displays tank temperatures in three different heights and the control signal as given by the CTRHYS block.



Please observe that the feedback loop is not algebraic because the TANKST block is a Delay block.

4.6 Block CTRL4DY

The CTRL4DY block simulates a real controler of a solar system of a solar driven absorption cooling machine including DHW and heating support.



Name	CTRL4DY
Function	st0039
Inputs	12
Outputs	11
Parameters	11
Strings	0
Group	S

Inputs

1 Collector temperature / °C
2 Collector outlet temperature at HX / °C
3 Storage tank temperature at top node / °C
4 Storage tank temperature at bottem node / °C
5 Time in seconds / s
6 DHW storage tank temperature at top node / °C
7 DHW storage tank temperature at bottom node / °C
8 Required mass flow DHW / kg s^{-1}
9 Inlet temperature of cold water / kg s^{-1}
10 Heating load DHW / kW
11 Heating load building / kW
12 Required supply temperature heating system / °C

Outputs

1 Controlled collector mass flow primary pump / kg s^{-1}
2 Controlled collector mass flow secondary pump / kg s^{-1}
3 Electricity consumption primary collector pump / kW
4 Electricity consumption secondary collector pump / kW

5 Heating energy transferred to storage tank / kW
6 Charge mass flow DHW tank / kg s^{-1}
7 Load mass flow DHW tank / kg
8 Heating load DHW not covered / kW
9 Mass flow to heating system / kg s^{-1}
10 Return temperature from heating circuit / °C
11 Heating energy not delivered by solar system / kW

Parameters

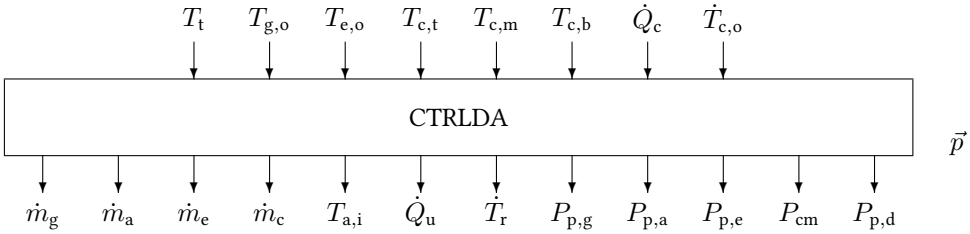
1 Nominal collector mass flow primary pump / kg s^{-1}
2 Nominal collector mass flow secondary pump / kg s^{-1}
3 Temperature hysteresis / K
4 Minimum operation time / s
5 Electricity consumption primary collector pump / kW
6 Electricity consumption secondary collector pump / kW
7 Control options for the collector pumps 0 - separate control primary and secondary pump 1 - common control primary and secondary pump
8 Charge mass flow DHW tank / kg s^{-1}
9 Supply water temperature / °C
10 Heat capacity of water / kJ kg^{-1} K^{-1}
11 Temperature difference building heating circuit / K

Strings

None

4.7 Block CTRLDA

The CTRLDA block simulates a real controler of a solar driven double effect absorption cooling machine.



Name	CTRLDA
Function	st0040
Inputs	8
Outputs	12
Parameters	14
Strings	0
Group	S

Inputs

1 Hot water storage temperature at top node / °C
2 Generator outlet temperature / °C
3 Evparator outlet temperature / °C
4 Cold water temperature at top node / °C
5 Cold water temperature at middle node / °C
6 Cold water temperature at bottom node / °C
7 Cooling load / kW
8 Outlet temperature cooling tower / °C

Outputs

1 Controlled generator mass flow / kg s^{-1}
2 Controlled absorber mass flow / kg s^{-1}
3 Controlled evaporator mass flow / kg s^{-1}
4 Controlled mass flow cold distribution taken from cold storage / kg s^{-1}
5 Contolled absorber inlet temperature / °C
6 Not delivered cooling power / kW
7 Return temperature from load / °C
8 Electricity consumption generator pump / kW

9	Electricity consumption absorber pump / kW
10	Electricity consumption evaporator pump / kW
11	Electricity consumption ACM / kW
12	Electricity consumption cold distribution pump / kW

Parameters

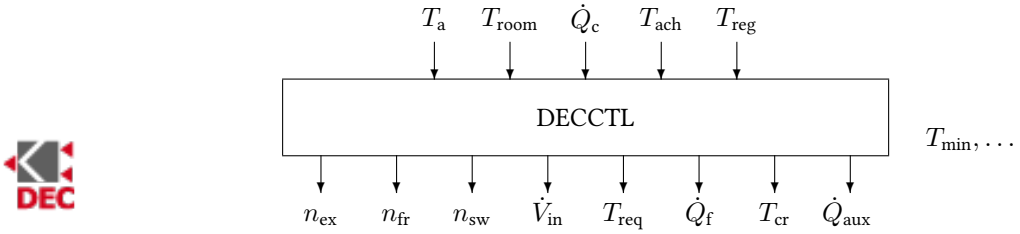
1	Generator mass flow / $kg\,s^{-1}$
2	Absorber mass flow / $kg\,s^{-1}$
3	Evaporator mass flow / $kg\,s^{-1}$
4	Minimum generator inlet temperature / °C
5	Generator minimum start up temperature / °C
6	Absorber inlet set temperature / °C
7	Cold distribution, supply temperature / °C
8	Cold distribution, return temperature / °C
9	Lower limit evaporator outlet temperature / °C
10	Electricity consumption generator pump / kW
11	Electricity consumption absorber pump / kW
12	Electricity consumption evaporator pump / kW
13	Electricity consumption ACM / kW
14	Electricity consumption cold distribution pump / kW

Strings

None

4.8 Block DECCTL

The DECCTL block controls the components of a desiccant cooling unit as a function of the required cooling load.



Name	DECCTL
Function	st0022
Inputs	5
Outputs	8
Parameters	10
Strings	0
Group	L

Inputs

1 Ambient temperature T_a / °C
2 Room temperature T_{room} / °C
3 Cooling load $\dot{Q}_c$ / kW
4 Achieved inlet air temperature as a result of the control strategy T_{ach} / °C
5 Regeneration air temperature T_{reg} / °C

Parameters

1 Minimum room inlet air temperature / °C
2 Minimum regeneration temperature / °C
3 Maximum regeneration temperature / °C
4 Nominal volume flow of fresh air ventilator / $m^3\,h^{-1}$
5 Nominal volume flow of regeneration air ventilator / $m^3\,h^{-1}$
6 Nominal power of auxiliary heating / kW
7 Auxiliary heating temperature accuracy / °C
8 Maximum number N_{max} of iterations
9 Normalised minimum air flow rate
10 Control strategy: 0 = electricity optimised; 1 = water consumption optimised

Outputs

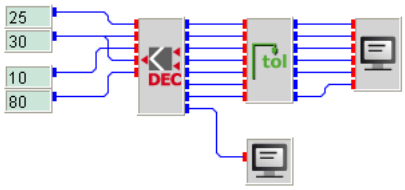
1	Status of exhaust air humidifier n_{ex}
2	Status of inlet air humidifier (stage 1 to 3) n_{fr}
3	Status of sorption wheel n_{sw}
4	Normalised volume flow rate of inlet air $\dot{V}_{in} \in [0, 1]$
5	Required air temperature T_{req} / °C
6	Free cooling power $\dot{Q}_f$ / kW
7	Controlled regeneration temperature T_{cr} / °C
8	Auxiliary heating power $\dot{Q}_{aux}$ / kW

Strings

None

---

decctl.vseit



The main input of the DECCTL block is the achieved inlet air temperature (input 4) which is usually the result of the control stratgy of the DECCTL block itself and its corresponding TOL block.

In order to demonstrate the control behaviour of the DECCTL block this input remains unchanged as defined by a CONST block. In other words, the controller will switch on all units of the desiccant cooling systems which are available.

All other inputs, like ambient temperature etc are set constant as well: ambient temperature 25 °C, room temperature 30 °C, cooling load 10 kW, regeneration air temperature 80 °C.

The parameters of the DECCTL block are set to these values:

```
Minimum room inlet air temperature                    12
Minimum regeneration temperature                      45
Maximum regeneration temperature                     120
Nominal volume flow of fresh air ventilator         5000
Nominal volume flow of regeneration air ventilator  5000
Nominal power of auxiliary heating                  1000
Auxiliary heating temperature accuracy                0.1
Maximum number of iterations                        100
Normalized minimum air flow rate                    1500
Control strategy                          0 = electricity optimized
```

Since one SCREEN block is directly connected to the TOL block it displays the values of all iteration steps while the second SCREEN block is called only after the iteration has stopped.

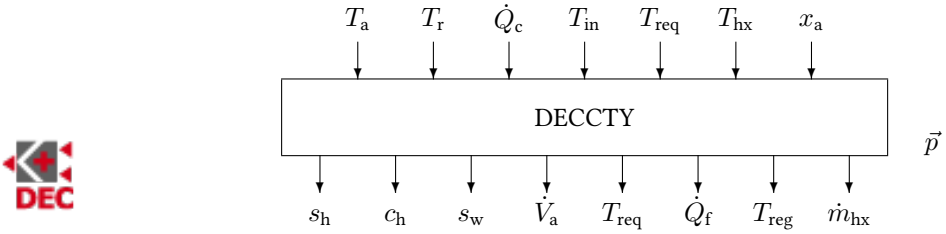
```
  0.0    0.0    0.0    0.0     0.0     0.0
  0.0    0.0    0.0    0.0    30.0    80.0
  0.0    0.0    0.0    0.3    12.0    80.0
  1.0    0.0    0.0    0.3    12.0    80.0
  1.0    1.0    0.0    0.3    12.0    80.0
  1.0    2.0    0.0    0.3    12.0    80.0
  1.0    3.0    0.0    0.3    12.0    80.0
  1.0    3.0    1.0    0.3    12.0    80.0
  0.0    0.0    0.0    1.0    24.0    80.0
  1.0    0.0    0.0    1.0    24.0    80.0
  1.0    1.0    0.0    1.0    24.0    80.0
  1.0    2.0    0.0    1.0    24.0    80.0
  1.0    3.0    0.0    1.0    24.0    80.0
  1.0    3.0    1.0    1.0    24.0    80.0
  1.0    3.0    1.0    1.0    24.0    80.0
  1.0    3.0    1.0    1.0    24.0   120.0
67.140
```

On initialization all outputs of the DECCTL block (and therefore also the outputs of the TOL block) are set to zero. In the first iteration the air flow of the desiccant cooling unit is switched on by 30 % and the required room temperature to achieve the cooling load demand of the given 10 kW is set to 12 °C.

In the next step the exhaust air humidifier is switched on. All options are switched on one after the other. As a result an auxiliary heating power of 67.140 kW would be required.

4.9 Block DECCTY

The DECCTY block controls the components of a desiccant cooling unit as a function of the required cooling load optimized.



Name	DECCTY
Function	st0038
Inputs	7
Outputs	8
Parameters	10
Strings	0
Group	L

Inputs

1 Ambient temperature / °C
2 Room temperature / °C
3 Cooling load / kW
4 Achieved inlet air temperature as a result of the control strategy / °C
5 Regeneration air temperature / °C
6 Regeneration air after HX / °C
7 Absolut humidity ambient air / g kg^{-1}

Parameters

1 Minimum room inlet air temperature / °C
2 Maximum regeneration temperature / °C
3 Minimum regeneration temperature / °C
4 Nominal volume flow of fresh air ventilator / m^3 h^{-1}
5 Minumum volume flow of fresh air ventilator / m^3 h^{-1}
6 Nominal volume flow of regeneration air ventilator / m^3 h^{-1}
7 Nominal power of auxiliary heating / kW
8 Auxiliary heating temperature accuracy/ °C
9 Maximum number N_{max} of iterations

10	Type of supply air humidifier: 0 contact matrix; 1 spray

Outputs

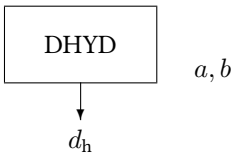
1	Status of exhaust air humidifier
2	Control signal inlet air humidifier (0.2 ...1.0)
3	Status of sorption wheel
4	Volume flow rate of inlet air (0 . . . 1)
5	Required air temperature / °C
6	Free cooling power / kW
7	Controlled regeneration temperature / °C
8	Controlled mass flow solar HX / kg s^{-1}

Strings

None

4.10 Block DHYD

The DHYD block calculates the hydraulic diameter of a rectangular channel.



Name	DHYD
Function	st0016
Inputs	0
Outputs	1
Parameters	2
Strings	0
Group	C

Inputs

None

Outputs

1	Hydraulic diameter d_h / m

Parameters

1	Channel width a / m
2	Channel height b / m

Strings

None

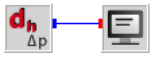
Description

The hydraulic diameter d_h is defined as the diameter of a concentric tube which has the same pressure drop per length as a channel with rectangular cross section when the velocity of the air flux is equal to the velocity in the channel with rectangular cross section. The hydraulic diameter is calculated by

$$d_\mathrm{h} = \frac{2ab}{a+b}$$

The side ratio of the rectangle should not exceed 1:5.

dhyd.vseit

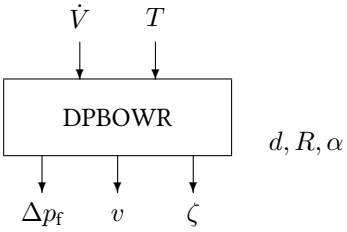


The DHYD block calculates the hydraulic diameter for a channel width of 0.3 m and a height of 0.2 m. The SCREEN block displays the result

```
0.240000
```

4.11 Block DPBOWR

The DPBOWR block calculates the pressure drop of a round bow due to friction of an incompressible one-phase turbulent tubular flow.



Name	DPBOWR
Function	st0021
Inputs	2
Outputs	3
Parameters	3
Strings	0
Group	S

Inputs

1	Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2	Air temperature T / °C

Outputs

1	Pressure drop Δp_f / Pa
2	Flow velocity v / $m\,s^{-1}$
3	Zeta value ζ

Parameters

1	Tube diameter d / m
2	Bow radius R / m
3	Angle α / °

Strings

None

Description

If the angle of the knee differs from $90°$ the ζ value can be approximated by

$$\zeta = \zeta_{90} k_\alpha$$

where ζ_{90} denotes the ζ value of a $90°$ knee. According to Idelchick the correction factor k_α satisfies

$$k_\alpha = \begin{cases} 0.9 \sin(\alpha) & \alpha \leq 60° \\ 0.0646\, \alpha^{0.609} & 60° < \alpha < 110° \\ 0.7 + 0.35\, \alpha / 90° & \alpha \geq 110° \end{cases}$$

dpbowr.vseit

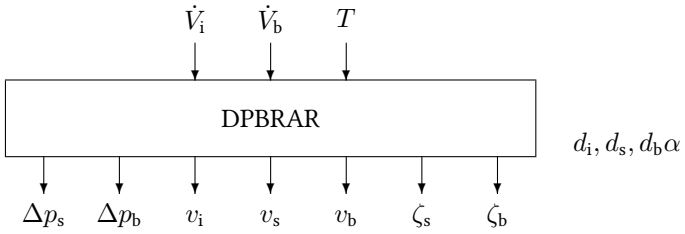
The DPBOWR block calculates the pressure drop, flow velocity and zeta value for a volume flow of 1800 m^3/h and an air temperature of 20 °C as defined through two CONST blocks.

The result is displayed by a SCREEN block.

```
1.31784     3.97887     0.14012
```

4.12 Block DPBRAR

The DPBRAR block calculates the pressure drop of a round branch.



Name	DPBRAR
Function	st0023
Inputs	3
Outputs	3
Parameters	4
Strings	0
Group	S

Inputs

1 Inlet volume flow $\dot{V}_\mathrm{i}$ / $\mathrm{m^3\,h^{-1}}$
2 Branch volume flow $\dot{V}_\mathrm{b}$ / $\mathrm{m^3\,h^{-1}}$
3 Air temperature T / °C

Outputs

1 Pressure drop Δp_s / Pa
2 Pressure drop Δp_b / Pa
3 Flow velocity v_i / $\mathrm{m\,s^{-1}}$
4 Flow velocity v_s / $\mathrm{m\,s^{-1}}$
5 Flow velocity v_b / $\mathrm{m\,s^{-1}}$
6 Zeta value ζ_s
7 Zeta value ζ_b

Parameters

1 Inlet tube diameter d_i / m
2 Straight tube diameter d_s / m
3 Branch tube diameter d_b / m
4 Branch angle α / °

Strings

None

Description

In air-conditioning it is common practise to separate the pressure drop in branches into two parts: One for the straight tube outlet and another one for the branch tube outlet. The advantage is that the two pressure drops then may be assigned to the two connected part channels.

The ζ value for the straight connection is found from

$$\zeta_s = k_c \left(1 - \frac{v_i}{v_s}\right)^2$$

where v_i denotes the flow velocity of the inlet air into the straight tube, and v_s the velocity out of the straight tube. The constant k_c is approximated by 0.4. The pressure drop is then equal to

$$\Delta p_s = \zeta_s \frac{\rho}{2} {v_s}^2$$

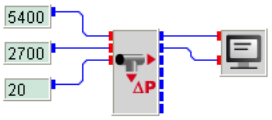
The ζ value for the branch connection is found from

$$\zeta_b = k_a \left(1 + \left(\frac{v_s}{v_b}\right)^2 - 2 \frac{v_s}{v_b} \cos(\alpha)\right)$$

where v_b denotes the flow velocity out of the branch tube. The constant k is approximated by 0.9 for $v_b/v_s > 0.8$ and $k = 1$ else. The pressure drop is then equal to

$$\Delta p_b = \zeta_b \frac{\rho}{2} {v_b}^2$$

dpbrar.vseit



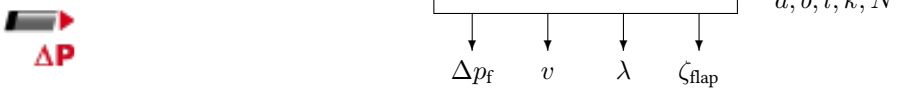
The inlet volume flow of the DPBRAR block is set to 5400 m^3/h assumed equally distributed to the straight tube and the branch tube as defined by two CONST blocks. The air temperature is 20 °C as defined through the third CONST block.

The SCREEN block displays the pressure drops in the two branches in Pascal.

```
5.942  74.273
```

4.13 Block DPCH

The DPCH block calculates the pressure drop in a rectangular channel due to friction of an incompressible one-phase turbulent tubular flow from the empirical Colebrook-White equation and the hydraulic diameter. It is possible to include a flap in the channel.



Name	DPCH
Function	st0007
Inputs	2 . . . [3]
Outputs	4
Parameters	4 . . . [5]
Strings	0
Group	S

Inputs

1 Volume flow $\dot{V}$ / $\mathrm{m^3\,h^{-1}}$
2 Air temperature T / °C
3 Flap angle δ / ° (Open: $\delta = 0°$. Closed: $\delta = 90°$)

Outputs

1 Pressure drop Δp_f / Pa
2 Flow velocity v / $\mathrm{m\,s^{-1}}$
3 Tube friction number λ / $\mathrm{m\,s^{-1}}$
4 Zeta value of flap position ζ_flap

Parameters

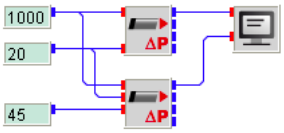
1 Side length a / m
2 Side length b / m
3 Length l / m
4 Channel roughness k / mm
5 Number of flaps N (either zero, or up to three)

Strings

None

---

dpch.vseit



Two rectangular air channels with side lengths 0.3 and 0.2 m and a length of 1 m are simulated for a volume flow of 1000 m^3/h and air temperature 20 °C as defined through two CONST blocks.

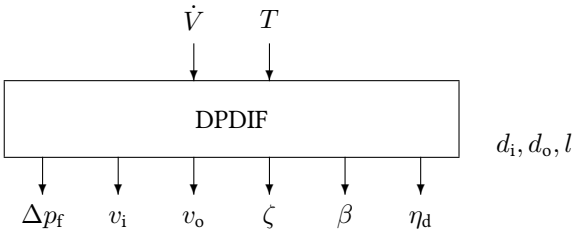
The upper DPCH block has no flaps while the lower block is assumed to have one flap at an opening angle of 45 ° as set by a third CONST block. The air temperature is set to 25 °C.

The SCREEN block displays the pressure drops in Pascal.

```
1.14992 219.52522
```

4.14 Block DPDIF

The DPDIF block calculates the pressure drop in a diffuser due to friction of an incompressible one-phase turbulent tubular flow.



Name	DPDIF
Function	st0018
Inputs	2
Outputs	6
Parameters	3
Strings	0
Group	S

Inputs

1 Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2 Air temperature T / °C

Outputs

1 Pressure drop Δp_f / Pa
2 Inlet flow velocity v_i / $m\,s^{-1}$
3 Outlet flow velocity v_o / $m\,s^{-1}$
4 Zeta value ζ
5 Opening angle β / °
6 Diffuser efficiency η_d

Parameters

1 Inlet diameter d_i / m
2 Outlet diameter d_o / m
3 Length l / m

Strings

None

Description

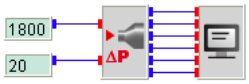
The ζ value of a diffuser can be found from

$$\zeta = \left(1 - (\frac{v_2}{v_1})^2\right)(1 - \eta_d) = \left(1 - (\frac{A_1}{A_2})^2\right)(1 - \eta_d)$$

Values for the diffuser efficiency due Baumgarth/Hoerner/Reeker are given in the following table.

β	3	6	8	10	12	14	16	20	24	30	40
η_d	0.97	0.92	0.89	0.85	0.81	0.77	0.73	0.64	0.53	0.36	0.08

dpdif.vseit



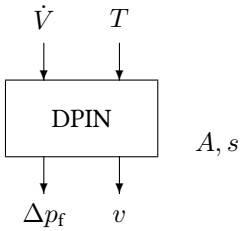
A diffuser with length 1 m is simulated with an inlet diameter of 0.25 m and an outlet diameter of 0.4 m. The volume flow is set to 1800 m^3/h at an air temperature of 20 °C as defined through two CONST blocks.

The SCREEN block displays all outputs of the DPDIF block.

```
6.34972  10.18592   3.97887   0.10302   8.57831   0.87843
```

4.15 Block DPIN

The DPIN block calculates the pressure drop from free space into a channel or tube.



Name	DPIN
Function	st0019
Inputs	2
Outputs	2
Parameters	2
Strings	0
Group	S

Inputs

1 Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2 Air temperature T / °C

Outputs

1 Pressure drop Δp_f / Pa
2 Flow velocity v / $m\,s^{-1}$

Parameters

1 Channel area A / m^2
2 Edge sharpness s
- 0 Sharp edge
- 1 Broken edge
- 2 Round edge
- 3 Smooth, round edge

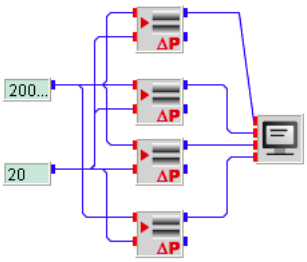
Strings

None

Description

The ζ value of sharp-edged inlets is assumed to depend on the shape of the inlet only. For sharp edges ζ is set to 0.35, for broken edges $\zeta = 0.11$, round edges $\zeta = 0.01$, for smooth, round edges ζ is set to zero.

dpin.vseit



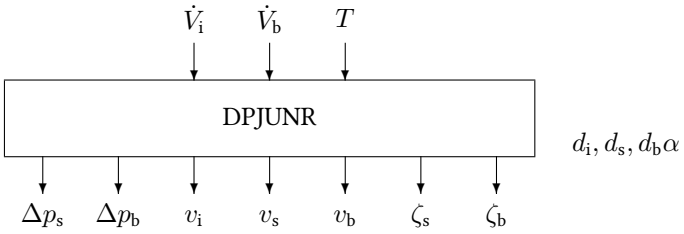
Four DPIN blocks with different edge sharpnesses are simulated at an air volume flow of 20 000 m^3/h at 20 °C as defined through two CONST blocks.

The SCREEN block displays the resulting pressure drops in Pascal.

```
6.41756   2.01695   0.18336   0.00183
```

4.16 Block DPJUNR

The DPJUNR block calculates the pressure drop of a round junction.



Name	DPJUNR
Function	st0023
Inputs	3
Outputs	3
Parameters	4
Strings	0
Group	S

Inputs

1 Inlet volume flow $\dot{V}_\mathrm{i}$ / $\mathrm{m^3\,h^{-1}}$
2 Branch volume flow $\dot{V}_\mathrm{b}$ / $\mathrm{m^3\,h^{-1}}$
3 Air temperature T / °C

Outputs

1 Pressure drop Δp_s / Pa
2 Pressure drop Δp_b / Pa
3 Flow velocity v_i / $\mathrm{m\,s^{-1}}$
4 Flow velocity v_s / $\mathrm{m\,s^{-1}}$
5 Flow velocity v_b / $\mathrm{m\,s^{-1}}$
6 Zeta value ζ_s
7 Zeta value ζ_b

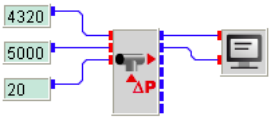
Parameters

1 Inlet tube diameter d_i / m
2 Straight tube diameter d_s / m
3 Branch tube diameter d_b / m
4 Branch angle α / °

Strings

None

dpjunr.vseit



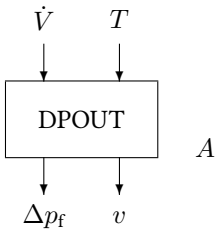
A round junction with inlet and straight tube diameter 0.9098 m and a branch tube diameter of 0.2764 m is simulated for an inlet volume flow of 4320 m^3/hand a branch volume flow of 5000 m^3/h of air at a temperature of 20 °C, as defined through three CONST blocks.

The resulting pressure drops in Pascal are displayed by a SCREEN block.

```
98.187    261.715
```

4.17 Block DPOUT

The DPOUT block calculates the pressure drop for a channel or tube outlet into free space.



Name	DPOUT
Function	st0019
Inputs	2
Outputs	2
Parameters	1
Strings	0
Group	S

Inputs

1 Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2 Air temperature T / °C

Outputs

1 Pressure drop Δp_f / Pa
2 Flow velocity v / $m\,s^{-1}$

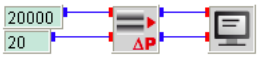
Parameters

1 Channel area A / m^2

Strings

None

dpout.vseit



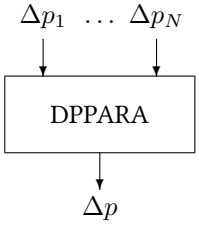
A channel or tube outlet with a channel area of 1 m^2 is simulated at an air volume flow of 20 000 m^3/h at a temperature of 20 °C, as defined through two CONST blocks.

A SCREEN block displays the resulting pressure drop in Pascal and the flow velocity in m/s.

```
18.335884  5.555555
```

4.18 Block DPPARA

The DPPARA block calculates the pressure drop of channels connected in parallel.



Name	DPPARA
Function	st0024
Inputs	2 . . . [10]
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1	Pressure drop of channel 1 Δp_1 / Pa
2	Pressure drop of channel 2 Δp_2 / Pa
N	Pressure drop of channel N Δp_N / Pa

Outputs

1	Overall pressure drop Δp / Pa

Parameters

None

Strings

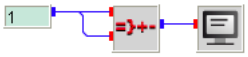
None

Description

The overall pressure drop Δp is calculated from the parallel connected input pressure drops $\Delta p_1, \Delta p_2, \ldots \Delta p_N$ by

$$\frac{1}{\Delta p} = \sum_{i=1}^{N} \frac{1}{\Delta p_i}$$

dppara.vseit

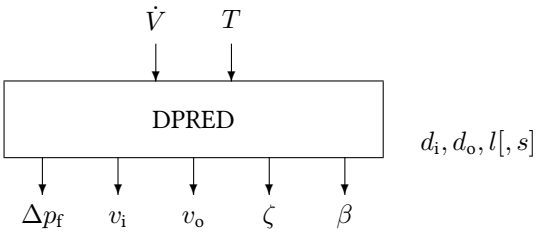


The most trivial application of the DPPARA block is that two channels with a pressure drop of 1 Pa each are connected in parallel so that the resulting pressure drop becomes 0.5 Pa as displayed by the SCREEN block.

```
0.50
```

4.19 Block DPRED

The DPRED block calculates the pressure drop in a reducer due to friction of an incompressible one-phase turbulent tubular flow.



Name	DPRED
Function	st0020
Inputs	2
Outputs	5
Parameters	3 . . . [4]
Strings	0
Group	S

Inputs

1 Volume flow $\dot{V}$ / $\mathrm{m^3\,h^{-1}}$
2 Air temperature T / °C

Outputs

1 Pressure drop Δp_f / Pa
2 Inlet flow velocity v_i / $\mathrm{m\,s^{-1}}$
3 Outlet flow velocity v_o / $\mathrm{m\,s^{-1}}$
4 Zeta value ζ
5 Closing angle β / °

Parameters

1 Inlet diameter d_i / m
2 Outlet diameter d_o / m
3 Length l / m
4 Sharpness s (evaluated only if BP(3) = 0)
  0 Sharp edge
  1 Broken edge
  2 Round edge

3 Smooth, round edge

Strings

None

Description

For the reducer no analytical equation is available in the current version. An empirical table is used instead. Since the ζ values are rather small, no interpolation between the values is made.

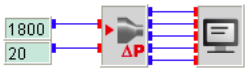
	A_o/A_i									
$\beta/°$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
≤ 10	0	0	0	0	0	0	0	0	0	0
20	0.01	0.01	0.01	0.01	0.01	0.01	0	0	0	0
30	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0
40	0.03	0.03	0.03	0.03	0.02	0.02	0.02	0.01	0.01	0
50	0.05	0.05	0.05	0.04	0.04	0.03	0.03	0.02	0.01	0
60	0.07	0.07	0.06	0.06	0.05	0.04	0.04	0.03	0.01	0

For angles greater than 60° an abrupt change in diameter is assumed and the following equation is used for the calculation of the ζ value:

$$\zeta = \zeta'(1 - \frac{A_o}{A_i})$$

where ζ' dependes on the sharpness of the edge. For sharp edges ζ' is equal to 0.5, for broken edges ζ' = 0.2, for round edges ζ' = 0.12, while for smooth, round edges ζ' is equal to 0.03.

dpred.vseit



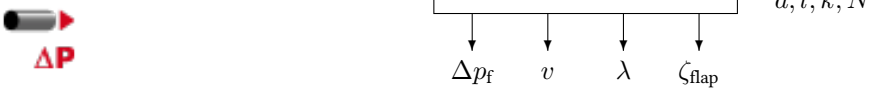
A reducer with length 1 m is simulated with an inlet diameter of 0.4 m and an outlet diameter of 0.25 m. The volume flow is set to 1800 m^3/h at an air temperature of 20 °C as defined through two CONST blocks.

The SCREEN block displays all outputs of the DPRED block for the sharpness “Broken edge”.

```
0.00616   3.97887  10.18592   0.00010   8.57831
```

4.20 Block DPTUBE

The DPTUBE block calculates the pressure drop in a tube due to friction of an incompressible one-phase turbulent tubular flow from the empirical Colebrook-White equation. It is possible to include a flap in the tube.



Name	DPTUBE
Function	st0007
Inputs	2 . . . [3]
Outputs	4
Parameters	3 . . . [4]
Strings	0
Group	S

Inputs

1 Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2 Air temperature T / °C
3 Flap angle δ / ° (Open: $\delta = 0°$. Closed: $\delta = 90°$)

Outputs

1 Pressure drop Δp_f / Pa
2 Flow velocity v / $m\,s^{-1}$
3 Tube friction number λ / $m\,s^{-1}$
4 Zeta value of flap position ζ_{flap}

Parameters

1 Nominal tube diameter d / m
2 Length l / m
3 Tube roughness k / mm
4 Number of flaps N (either zero or one)

Strings

None

Tube type	Roughness k / mm
Smooth tubes (brass etc.)	0.0015
PVC and PE tubes	0.007
Asbestos-cement tubes (new)	0.05 . . . 0.1
Steel tubes	0.045
Galvanised steel tubes	0.15
Steel tubes, light rust	0.15 . . . 1.0
Steel tubes, heavy rust	1.0 . . . 3.0
Cast iron tubes	0.4 . . . 0.6
Cast iron tubes, asphaltic	0.125
Sheet metal channel, weltet	0.15
Flexible tubes	0.6 . . . 0.8 . . . > 2.0
Stonewalled channels	3.0 . . . 5.0
Wooden channels	0.2 . . . 1.0
Concrete channels, rough	1.0 . . . 3.0

Table 4.1: Surface roughness of different tubes according to Recknagel.

Description

The empirical Colebrook-White equation is given by

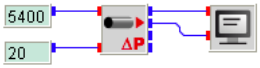
$$\frac{1}{\lambda} = -2 \log \left(\frac{2.51}{\mathrm{Re}\sqrt{\lambda}} + \frac{k}{3.71\, d} \right)$$

where λ denotes the tube friction coefficient, k the surface roughness of the tube, d the nominal tube diameter, and Re the Reynold's number $\mathrm{Re} = vd/\nu$ with the flow velocity v and the viscosity of air ν. Hence, λ is a function of the Reynold's number and the relative surface rougness $\epsilon = k/d$. Some typical k-values are given in Table.

The pressure drop as function of tube length L and air density ρ is given by

$$\Delta p = \lambda \frac{L}{d} \rho \frac{v^2}{2}$$

dptube.vseit

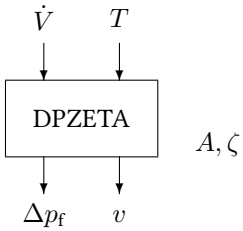


A tube with length 1 m and a diameter of 0.35 m is simulated with a tube roughness set to 0.15. No flap is assumed. The volume flow is set to 5400 m^3/h at an air temperature of 20 °C as defined through two CONST blocks. The SCREEN block displays the pressure drop and the flow velocity.

```
7.241074 15.590689
```

4.21 Block DPZETA

The DPZETA block calculates the pressure drop for a given zeta value.



Name	DPZETA
Function	st0019
Inputs	2
Outputs	2
Parameters	2
Strings	0
Group	S

Inputs

1 Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2 Air temperature T / °C

Outputs

1 Pressure drop Δp_f / Pa
2 Flow velocity v / $m\,s^{-1}$

Parameters

1 Channel area A / m^2
2 Zeta value ζ

Strings

None

Description

For a given ζ value the pressure drop is given by

$$\Delta p_f = \zeta \frac{\rho_{air}}{2v^2}$$

where v is the flow velocity.

dpzeta.vseit

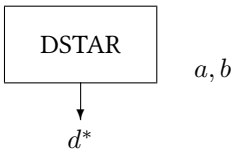
The DPZETA block calculates the pressure drop for a channel with an area of 1 m^2 and a zeta value of 0.3. The volume flow is set to 10 000 m^3/h for air at a temperature of 20 °C as set by two CONST blocks.

The SCREEN block displays the pressure drop in Pa and the flow velocity in m/s..

```
1.3751915       2.7777777
```

4.22 Block DSTAR

The DSTAR block calculates the equivalent diameter of a rectangular channel.



Name	DSTAR
Function	st0016
Inputs	0
Outputs	1
Parameters	2
Strings	0
Group	C

Inputs

None

Outputs

1 Equivalent diameter d^* / m

Parameters

1 Channel width a / m
2 Channel height b / m

Strings

None

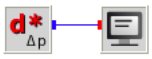
Description

The equivalent diameter d^* is defined as the diameter of a concentric tube which has the same pressure drop per length as a channel with rectangular cross section when the volume flow of the air flux is equal to the volume flow in the channel with rectangular cross section. The equivalent diameter is calculated by

$$d^* = 1.27 \sqrt[5]{\frac{a^3\, b^3}{a+b}}$$

The side ratio of the rectangle should not exceed 1:5.

dstar.vseit



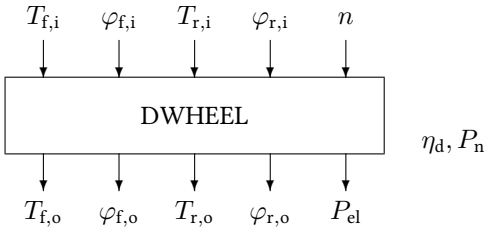
The DSTAR block calculates the equivalent diameter for a channel width of 0.3 m and a channel height of 0.2 m.

The SCREEN block displays the result.

```
0.269712
```

4.23 Block DWHEEL

The DWHEEL block simulation a desiccant wheel.



Name	DWHEEL
Function	st0004
Inputs	5
Outputs	5
Parameters	2
Strings	0
Group	S

Inputs

1 Fresh air inlet temperature $T_{f,i}$ / °C
2 Fresh air inlet relative humidity $\varphi_{f,i}$
3 Regeneration air inlet temperature $T_{r,i}$ / °C
4 Regeneration air inlet relative humidity $\varphi_{r,i}$
5 Rotational speed n / r.p.m.

Outputs

1 Fresh air outlet temperature $T_{f,o}$ / °C
2 Fresh air outlet relative humidity $\varphi_{f,o}$
3 Regeneration air outlet temperature $T_{r,o}$ / °C
4 Regeneration air outlet relative humidity $\varphi_{r,o}$
5 Electricity consumption P_{el} / W

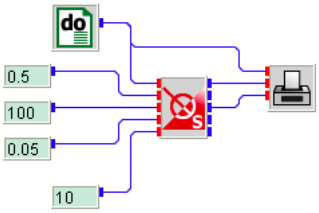
Parameters

1 Dehumidification efficiency η_d
2 Nominal electricity consumption P_n / W

Strings

None

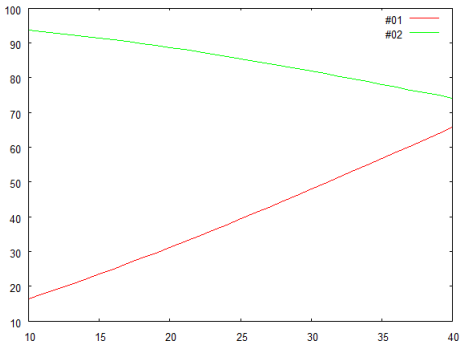
dwheel.vseit



A desiccant wheel with a dehumidification efficiency of 0.8 and a nominal electricity consumption of 50 W is simulated by the DWHEEL block.

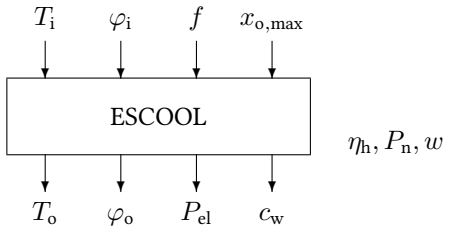
The fresh air inlet temperature is varied by a DO block between 10 and 40 °C in steps of 1 °C. The fresh air outlet temperature is plotted as a function of the fresh air inlet temperature together with the regeneration air outlet temperature by a PLOT block.

The other inputs of the DWHEEL block are defined through CONST blocks to a relative humidity of the fresh air of 50 percent, a regeneration air inlet temperature of 100 °C, a regeneration air inlet humidity of 5 percent, and a rotational speed of the rotor of 10 r.p.m.



4.24 Block ESCOOL

The ESCOOL block simulates an evaporative spray humidifier.



Name	ESCOOL
Function	st0035
Inputs	4
Outputs	4
Parameters	3
Strings	0
Group	S

Inputs

1 Temperature T_i /°C
2 Relative humidity φ_i
3 Control input f (0...100 %)
4 Maximum absolute humidity $x_\mathrm{o,max}$ / g kg^{-1}

Outputs

1 Outlet temperature T_o /°C
2 Relative outlet humidity φ_o
3 Electricity consumption P_el
4 Water consumption c_w/ g kg^{-1}

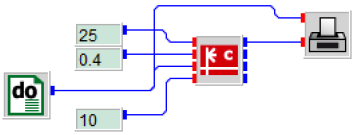
Parameters

1 Humidification maximum efficiency η_h
2 Nominal electricity consumption P_n / W
3 Water consumption factor w

Strings

None

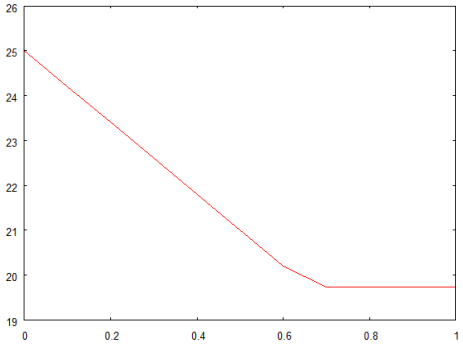
escool.vseit



A continuously drivable spray humidifier with a humidification efficiency of 0.9 and a nominal electricity consumption of 10 W is simulated by the ESCOOL block. The water consumption factor is set to 1.

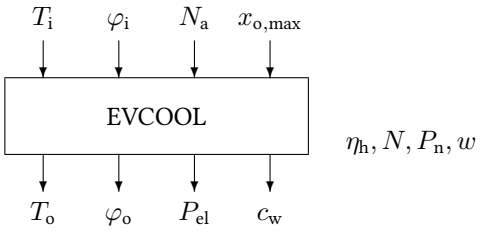
The control signal is varied by a DO block between 0 and 1 in steps of 0.1. The outlet temperature is plotted as a function of the control signal by a PLOT block.

The other inputs of the ESCOOL block are defined through CONST blocks to for an ambient temperature of 25 degrees Celsius at a relative humidity of the inlet air of 40 percent. The maximum absolute humidity is restricted to 10 kg water per kg air.



4.25 Block EVCOOL

The EVCOOL block simulates an evaporative cooler (humidifier).



Name	EVCOOL
Function	st0006
Inputs	4
Outputs	4
Parameters	4
Strings	0
Group	S

Inputs

1 Inlet temperature T_i / °C
2 Relative humidity φ_i
3 Number of active stages N_a
4 Maximum absolute humidity $x_\mathrm{o,max}$ / g kg^{-1}

Outputs

1 Outlet temperature T_o /°C
2 Relative outlet humidity φ_o
3 Electricity consumption P_el / W
4 Water consumption c_w/ g kg^{-1}

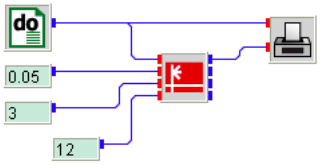
Parameters

1 Humidification efficiency η_h
2 Number of stages N
3 Nominal electricity consumption P_n / W
4 Water consumption factor w

Strings

None

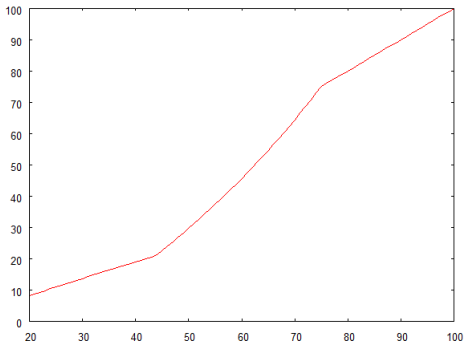
evcool.vseit



An three-stage evaporative cooler with a humidification efficiency of 0.9 and a nominal electricity consumption of 50 W is simulated by the EVCOOL block. The water consumption factor is set to 1.3.

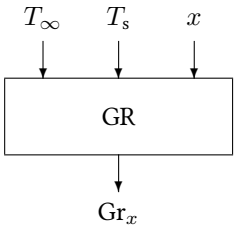
The inlet temperature is varied by a DO block between 20 and 100 °C in steps of 1 °C. The outlet temperature is plotted as a function of the inlet by a PLOT block.

The other inputs of the EVCOOL block are defined through CONST blocks to a relative humidity of the inlet air of 50 percent. The number of active stages is set to 3, the maximum absolute humidity is restricted to 12 kg water per kg air.



4.26 Block GR

The GR block calculates the Grashof number for air as an ideal gas.



Name	GR
Function	st0033
Inputs	3
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Fluid temperature T_∞ / °C
2 Surface temperature T_s / °C
3 Length x / °C

Outputs

1 Grashof number Gr_x

Parameters

None

Strings

None

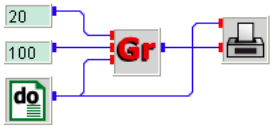
Description

The Grashof number is defined as

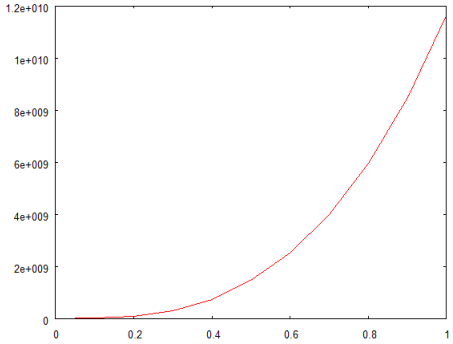
$$\mathrm{Gr}_x = \frac{g x^3 \beta \Delta T}{\nu^2}$$

where g is gravitational acceleration 9.81 $\mathrm{m\,s^{-2}}$, x is a characteristic length, $\beta = 1/T$ the thermal extension coefficient, ΔT the temperature difference between fluid and surface, and ν the kinematic viscosity.

gr.vseit

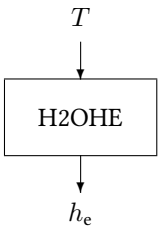


The Grashof number is plotted by a PLOT block as a function of lenghth, for a fluid temperature of 20 °C and a surface temperature of 100 °C, both set by CONST blocks.



4.27 Block H2OHE

The H2OHE block calculates the evaporation enthalpy of water as a function of temperature.



Name	H2OHE
Function	st0010
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Temperature T / °C

Outputs

1 Evaporation enthaply of water h_e / kJ kg^{-1}

Parameters

None

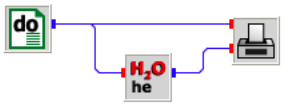
Strings

None

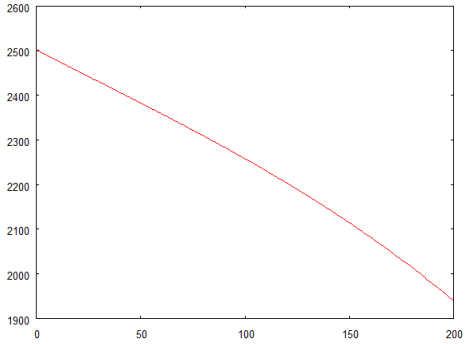
Description The temperature dependency of the the evaporation enthalpy is given as the difference between the enthalpy of vapor h_v and the enthalpy of the liquid h_l

$$h_e(T) = h_v(T) - h_l(T)$$

h2ohe.vseit



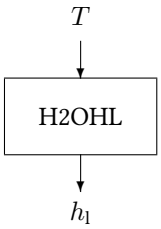
The PLOT block displays the evaporation enthalpy of water as a function of temperature which is varied between 0 and 200 °C in steps of 1 °C, as defined through a DO block.



See also Blocks H2OHL and H2OHV.

4.28 Block H2OHL

The H2OHL block calculates the enthalypy of water as a function of temperature.



Name	H2OHL
Function	st0010
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Temperature T / °C

Outputs

1 Enthaply of water h_l / kJ kg^{-1}

Parameters

None

Strings

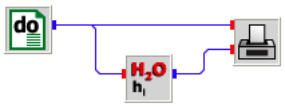
None

Description The enthalpy of liquid water h_l is calculated after the empirical correlation

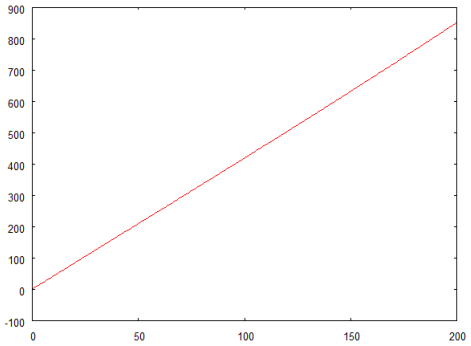
$$h_l = -\ 0.0225 + 4.2063437\ T - 6.014696 \cdot 10^{-4}\ T^2 + 4.381537 \cdot 10^{-6}\ T^3$$

The correlation is accurate within ± 0.04 % for a temperature range between 0 °C and 200 °C.

h2ohl.vseit



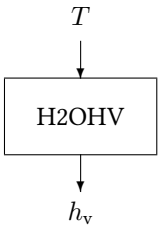
The PLOT block displays the enthalpy of liquid water as a function of temperature which is varied between 0 and 200 °C in steps of 1 °C, as defined through a DO block.



See also Blocks H2OHE and H2OHV.

4.29 Block H2OHV

The H2OHV block calculates the enthalpy of water vapor as a function of temperature.



Name	H2OHV
Function	st0010
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Temperature T / °C

Outputs

1 Enthaply of water vapor h_v / kJ kg^{-1}

Parameters

None

Strings

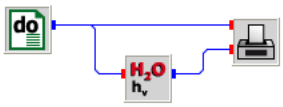
None

Description The enthalpy of water vapor h_v is calculated after the empirical correlation

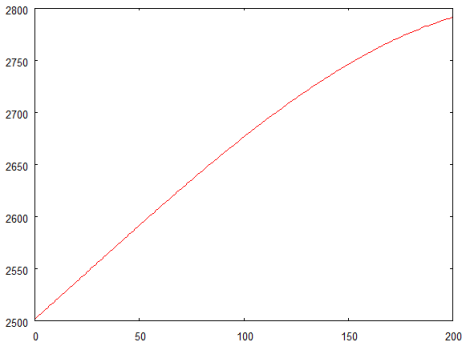
$$h_v = 2501.482 + 1.789736\,T + 8.957546 \cdot 10^{-4}\,T^2 - 1.300254 \cdot 10^{-5}\,T^3$$

The correlation is accurate within ± 0.04 % for a temperature range between 0 °C and 200 °C.

h2ohv.vseit



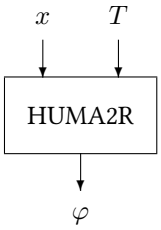
The PLOT block displays the enthalpy of water vapor as a function of temperature which is varied between 0 and 200 °C in steps of 1 °C, as defined through a DO block.



See also Blocks H2OHE and H2OHL.

4.30 Block HUMA2R

The HUMA2R block converts absolute humidity to relative humidity as a function of temperature.



Name	HUMA2R
Function	st0013
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Absolute humdity of humid air x / g water/kg dry air
2 Temperature T / °C

Outputs

1 Relative humdity of humid air φ

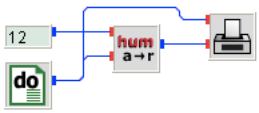
Parameters

None

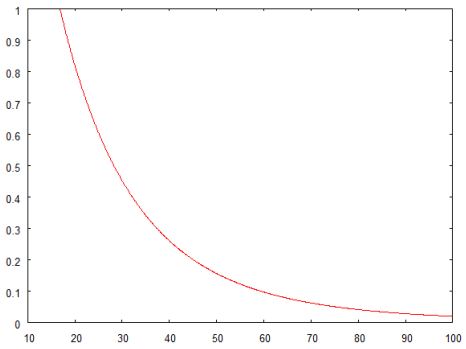
Strings

None

huma2r.vseit



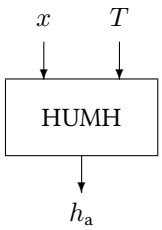
A DO block varies temperatures between 15 and 100 degrees Celsius. Assuming a constant absolute humidity of 12 grams of water per kilogram of air, the HUMA2R block calcultes the relative humidity which is displayed by the PLOT block as a function of temperature.



See also Block HUMR2A.

4.31 Block HUMH

The HUMH block returns the enthalpy of air as a function of absolute humidity and temperature.



Name	HUMH
Function	st0013
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Absolute humdity of humid air x / g water/kg dry air
2 Temperature T / °C

Outputs

1 Enthaply of air h_a / kJ kg^{-1}

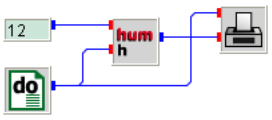
Parameters

None

Strings

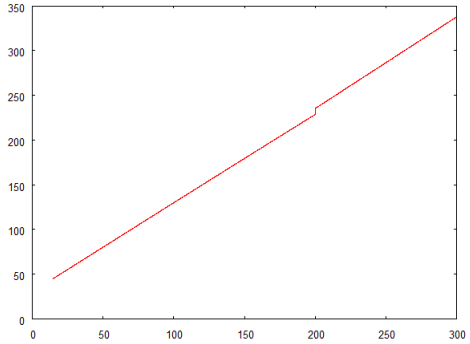
None

humh.vseit



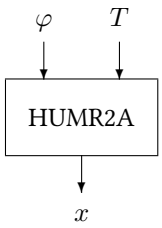
The PLOT block displays the enthalpy of humid air as a function of temperature which is varied between 15 and 300 °C in steps of 0.1 °C, as defined through a DO block.

The absolute humidity of the humid air is set to 12 g water per kg of dry air by a CONST block.



4.32 Block HUMR2A

The HUMR2A block converts relative humidity to absolute humidity as a function of temperature.



Name	HUMR2A
Function	st0013
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Relative humdity of humid air φ
2 Temperature T / °C

Outputs

1 Absolute humdity of humid air x / g water/kg dry air

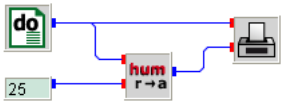
Parameters

None

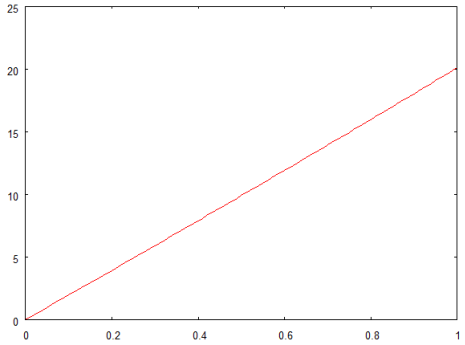
Strings

None

humr2a.vseit



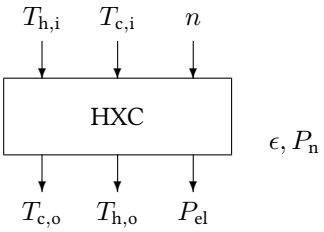
A DO block varies relative humidity from 0 to 1 in steps of 0.01. Assuming a constant temperature of 25 °C as defined by a CONST block the absolute humidity is displayed by a PLOT block.



See also Block HUMA2R.

4.33 Block HXC

The HXC block simulates a heat exchanger with constant heat recovery efficiency.



Name	HXC
Function	st0005
Inputs	3
Outputs	3
Parameters	2
Strings	0
Group	S

Inputs

1 Temperature (hot) $T_{h,i}$ / °C
2 Temperature (cold) $T_{c,i}$ / °C
3 On-Off switch 0/1

Outputs

1 Temperature (cold) $T_{c,o}$ / °C
2 Temperature (hot) $T_{h,o}$ / °C
3 Electricity consumption P_{el} / W

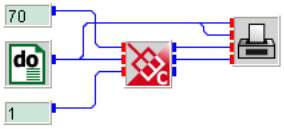
Parameters

1 Heat recovery efficiency ϵ
2 Nominal electricity consumption P_n / W

Strings

None

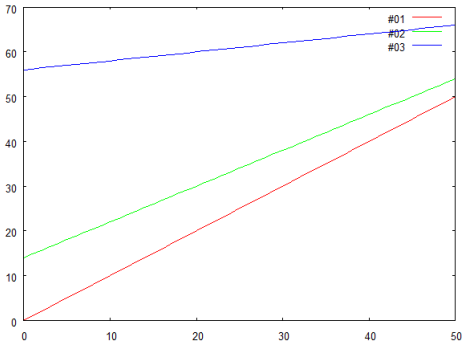
hxc.vseit



A heat exchanger with constant heat recovery efficiency of 80 % and a nominal electricity consumption of 50 W is simulated by a HXC block. The heat exchanger is switched on all the time and the hot side inlet temperature is set to 70 °C as defined by two CONST blocks.

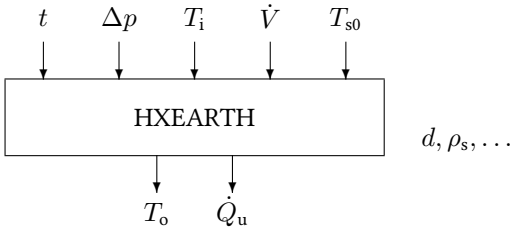
A DO block varies the cold side inlet temperature between 0 and 50 °C in steps of 1 °C.

The cold side inlet temperature and the resulting outlet temperatures are displayed by a PLOT block.



4.34 Block HXEARTH

The HXEARTH block calculates the outlet temperature of a part of a system of parallel horizontal groundheat collectors, taking into account the influence of the ground surface temperature and the groundwater level.



Name	HXEARTH
Function	st0031
Inputs	5
Outputs	2
Parameters	18
Strings	0
Group	S

Inputs

1 Time t / s
2 Pressure loss of the system Δp / Pa
3 Inlet air temperatur T_i / °C
4 Total mass flow $\dot{V}$ / m^3/s
5 Undisturbed soil temperature $T_{\mathrm{s}0}$ / °C

Outputs

1 Outlet air temperature T_o / °C
2 Useful heat $\dot{Q}_\mathrm{u}$ / W

Parameters

1 Depth of the tube below ground d / m
2 Soil density ρ_s / $\mathrm{kg\,m^{-3}}$
3 Heat capacity of the soil c_p / $\mathrm{J\,kg^{-1}\,K^{-1}}$
4 Heat conductivity of the soil λ / $\mathrm{W\,m^{-1}\,K^{-1}}$
5 Tube radius r / m
6 Total tube length l / m

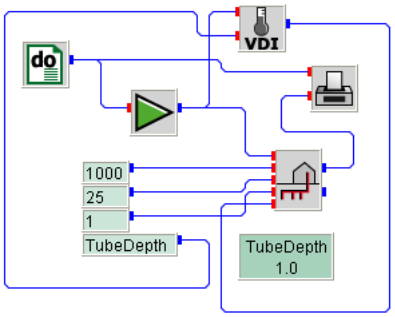
7 Fan efficiency η_F (usually between 0.3 and 0.9)
8 Fan position
  0 At entrance
  1 At exit of the tube
  2 No fan
9 Depth of influence of hourly temperature changes d_i / m (usually around 0.14 m)
10 Number of segments n
11 Depth of groundwater level d_w / m
12 Overall heat loss coefficient of building wall U / W m^{-2} K^{-1}
13 Air temperature inside the building T_b / °C
14 Average distance between heat collector and building Δ_b / m
15 Number of tubes in parallel N_p
16 Distance between centers of tubes Δx / m
17 Attenuation factor f
18 Ground temperature phase delay Δd /d

Strings

None

---

hxearth.vseit



In this example an earth heat exchanger is simulated with the following parameters:

Depth of the tube below ground	TubeDepth
Soil density	1900
Heat capacity of the soil	900
Heat conductivity of the soil	1.5
Tube radius	0.8
Total tube length	100
Fan efficiency	0.7
Fan position	At tube entrance
Depth of influence of hourly temperature changes	0.14
Number of segments	100
Depth of groundwater level	5
Overall heat loss coefficient of building wall	2.0
Air temperature inside the building	20

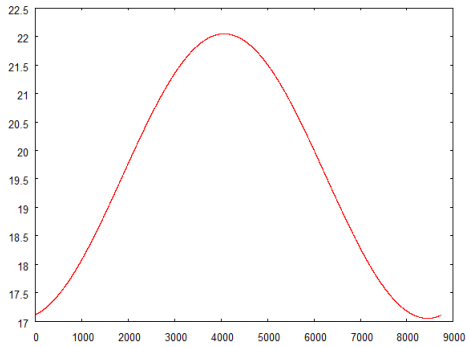
```
Average distance between heat collector and building      5
Number of tubes in parallel                               2
Distance between centers of tubes                         5
Attenuation factor                                      0.1
Ground temperature phase delay                        150
```

The Depth of the tube below ground parameter is set to the value TubeDepth which is a global variable defined to be 1.0, because this parameter is also required by the TSOIL block as input from a CONST block with its parameter set to TubeDepth as well.

Almost all inputs of the HXEARTH block are set constant through CONST blocks. The pressure loss of the system is 1000 Pa, the inlet air temperature is set to 25 °C, and the total air mass flow is 1 m^3/s.

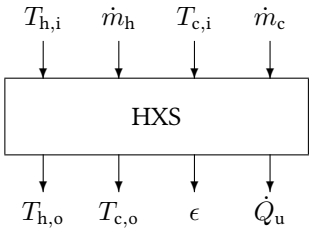
The required soil temperature is calculated by the TSOIL block after DIN 4710 for the region of Mannheim, German.

The simulated time period is 8760 hours as specified by a DO block. Since the HXEARTH block expects the time signal in seconds, a GAIN block is used to convert the hour signal of the DO block to the second of the year. A PLOT block displays the outlet air temperature of the earth heat exchanger as a function of the hour of the year.



4.35 Block HXS

The HXS block simulates a simple sensible heat exchanger.



mode,
$UA, c_{p,h}, c_{p,c}$

Name	HXS
Function	st0015
Inputs	4
Outputs	4
Parameters	4
Strings	0
Group	S

Inputs

1 Temperature $T_{h,i}$ / °C
2 Mass flow $\dot{m}_1$ / kg s^{-1}
3 Temperature $T_{c,i}$ / °C
4 Mass flow $\dot{m}_2$ / kg s^{-1}

Outputs

1 Temperature $T_{h,o}$ / °C
2 Temperature $T_{c,o}$ / °C
3 Effectiveness ϵ
4 Transfered heat $\dot{Q}_u$ / W

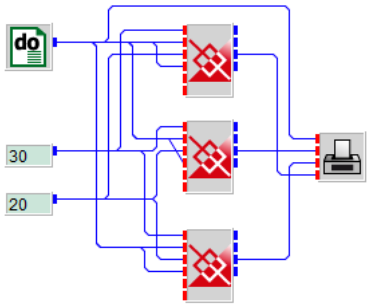
Parameters

1 Mode
  0 Parallel flow
  1 Counter flow
  2 Cross flow
2 Overall heat transfer coefficient UA / W K^{-1}
3 Specific heat of side 1 fluid $c_{p,h}$ / J kg^{-1} K^{-1}
4 Specific heat of side 2 fluid $c_{p,c}$ / J kg^{-1} K^{-1}

Strings

None

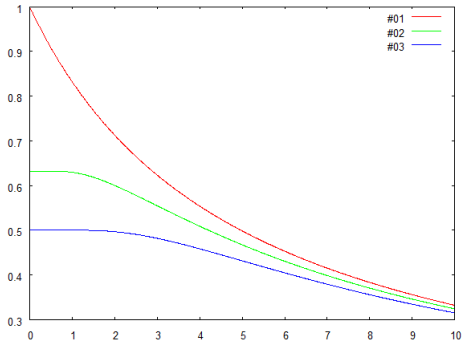
hxs.vseit



All three types of a simple heat exchangers are simulated: (i) parallel flow – upper one, (ii) counter flow – middle one, and (iii) cross flow – lower one. In all three cases the overall heat transfer coefficient is 5000 $W\,K^{-1}$, and both specific heat values of the fluid are set to 1007 $J\,kg^{-1}\,K^{-1}$ (air).

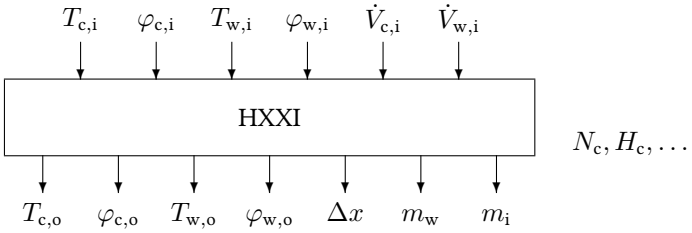
The inlet temperatures are set to 20 and 30 °C, respectively, defined by two CONST blocks.

A DO block varies the mass flow between 0.01 and 10 kg/s and a PLOT block is used to display the three different effectivenesses of the heat exchangers.



4.36 Block HXXI

The HXXI block simulates a cross-flow heat exchanger including condensation and icing on the heat exchanger walls.



Name	HXXI
Function	st0017
Inputs	6
Outputs	7
Parameters	17
Strings	0
Group	S

Inputs

1 Inlet temperature cold air stream $T_{c,i}$ / °C
2 Inlet relative humidity cold air stream $\varphi_{c,i}$
3 Inlet temperature warm air stream $T_{w,i}$ / °C
4 Inlet relative humidity warm air stream $\varphi_{w,i}$
5 Volume flow cold air stream $\dot{V}_{c,i}$ / $m^3\,s^{-1}$
6 Volume flow warm air stream $\dot{V}_{w,i}$ / $m^3\,s^{-1}$

Outputs

1 Outlet temperature cold air stream $T_{c,o}$ / °C
2 Outlet relative humidity cold air stream $\varphi_{c,o}$
3 Outlet temperature warm air stream $T_{w,o}$ / °C
4 Outlet relative humidity warm air stream $\varphi_{w,o}$
5 Dehumidification warm air stream by condensation Δx / $kg\,kg^{-1}$
6 Total condensation water mass m_w / $kg\,h^{-1}$
7 Total ice mass m_i / $kg\,h^{-1}$

Parameters

1 Number of air channels of the heat exchangers cold air side N_c

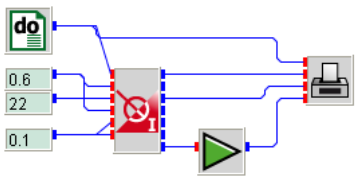
2 Channel height of the heat exchanger canals cold air side H_c / m
3 Channel width cold air side W_c / m
4 Channel length cold air side L_c / m
5 Number of air channels of the heat exchangers warm air side N_w
6 Channel height of the heat exchanger canals warm air side H_w / m
7 Channel width warm air side W_w / m
8 Channel lenght warm air side L_c / m
9 Thickness of the heat exchanger walls / m
10 Heat exchanger with ribs on the walls / 0=yes, 1=no
11 Thickness of the ribs / m
12 Number of ribs in each channel cold air side $N_{r,c}$
13 Number of ribs in each channel warm air side $N_{r,w}$
14 Heat conductivity of the heat exchanger walls λ / $\mathrm{W\,m^{-1}\,K^{-1}}$
15 Number of elements along I-Direction (division cold air channel width) N_I
16 Number of elements along J-Direction (division warm air channel width) N_J
17 Total pressure p / Pa (about 101300 Pa)

Strings

None

---

hxxi.vseit



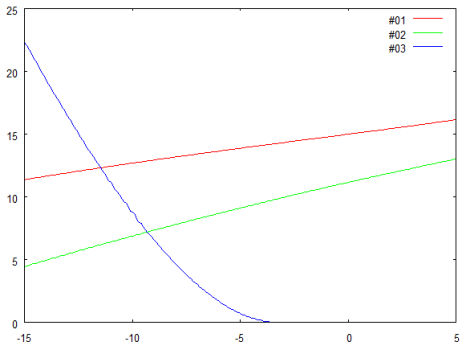
A cross flow heat exchanger with condensation and icing is simulated with the following parameters:

```
Number of air channels of the heat exchangers cold air side   35
Channel height of the heat exchanger canals cold air side      0.004
Channel width cold air side                                    0.5
Channel lenght cold air side                                   0.5
Number of air channels of the heat exchangers warm air side   35
Channel height of the heat exchanger canals warm air side      0.004
Channel width warm air side                                    0.5
Channel lenght warm air side                                   0.5
Thickness of the heat exchanger walls                          0.002
Heat exchanger with ribs on the walls                          0
Thickness of the ribs                                          0.004
Number of ribs in each channel cold air side                  30
Number of ribs in each canal warm air side                    30
Heat conductivity of the heat exchanger walls                  0.88
Number of elements along I-Direction                          50
Number of elements along J-Direction                          50
Total pressure                                            101300
```

A DO block varies the cold side air temperature of a heat exchanger between −15 and 5 °C in steps of 0.1 °C.

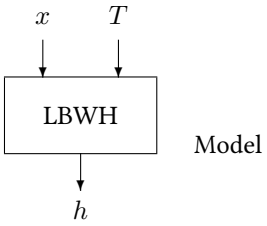
The other inputs of the HXXI block are defined through CONST blocks: Both relative humidities of the inlet air stream are set to 60 %, the warm inlet air stream temperature is set to 22 °C, and both volume flows are assumed to be 0.1 m^3/s.

The PLOT block displays the two outlet temperatures of the heat exchanger and the total ice mass in g/h as converterted from kg/h by a GAIN block with parameter 1000.



4.37 Block LBWH

The LBWH block returns the enthalpy of a lithium-bromide-water dilution as a function of concentration and temperature.



Model

Name	LBWH
Function	st0025
Inputs	2
Outputs	1
Parameters	2
Strings	0
Group	S

Inputs

1 Lithium-bromide concentration x / kg LiBr kg^{-1} H_2O
2 Temperature T / °C

Outputs

1 Enthaply of lithium-bromide-water dilution h / kJ kg^{-1}

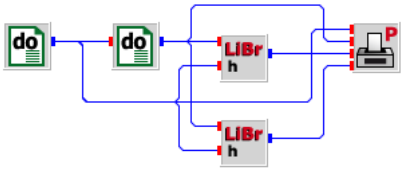
Parameters

1 Model
  0 Mc Neely 1979
  1 Kaita 1991
2 Outside-range value. If set to zero ($|BP(2)| < 10^{-5}$) data are extrapolated.

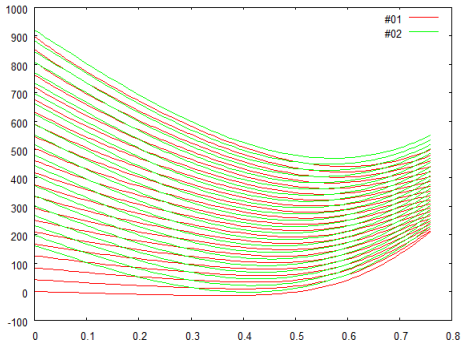
Strings

None

lbwh.vseit



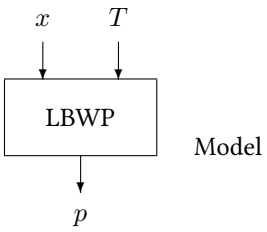
The enthalpy of lithium-bromide-water dilution is calculated after two different approximations and plotted by a PLOTP block as a function of concentration and temperature (McNeely: red, Kaita: green).



The concentration and temperature are varied by two nested DO blocks between 0 and 210 °C in steps of 10 °C (outer block) and 0 and 0.76 kg LiBr per kg water in steps of 0.01 (inner block).

4.38 Block LBWP

The LBWP block returns the pressure of a lithium-bromide-water dilution as a function of concentration and temperature.



Model

Name	LBWP
Function	st0025
Inputs	2
Outputs	1
Parameters	2
Strings	0
Group	S

Inputs

1 Lithium-bromide concentration x in / kg LiBr kg^{-1} H_2O
2 Temperature T / °C

Outputs

1 Pressure of lithium-bromide-water dilution p / Pa

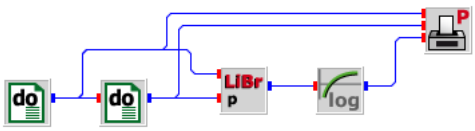
Parameters

1 Model
0 Mc Neely 1979
2 Outside-range value. If set to zero ($|BP(2)| < 10^{-5}$) data are extrapolated.

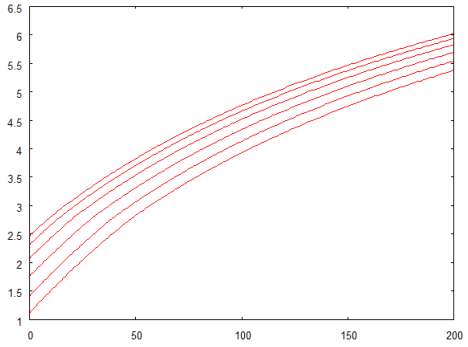
Strings

None

lbwp.vseit



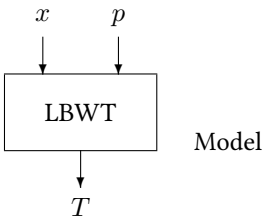
The pressure of lithium-bromide-water dilution is calculated and plotted by a PLOTP block as a function of concentration and temperature.



The concentration and temperature are varied by two nested DO blocks between 0 and 200 °C in steps of 1 °C (inner block) and 0.4 and 0.65 kg LiBr per kg water in steps of 0.05 (outer block).

4.39 Block LBWT

The LBWT block returns the temperature of a lithium-bromide-water dilution as a function of concentration and pressure.



Model

Name	LBWT
Function	st0025
Inputs	2
Outputs	1
Parameters	2
Strings	0
Group	S

Inputs

1 Lithium-bromide concentration x / kg LiBr kg^{-1} H_2O
2 Pressure p / Pa

Outputs

1 Temperature of lithium-bromide-water dilution T / °C

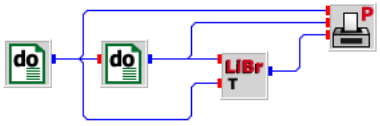
Parameters

1 Model
0 Mc Neely 1979
2 Outside-range value. If set to zero ($|BP(2)| < 10^{-5}$) data are extrapolated.

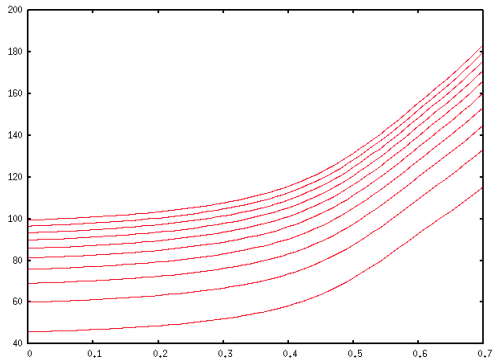
Strings

None

lbwt.vseit



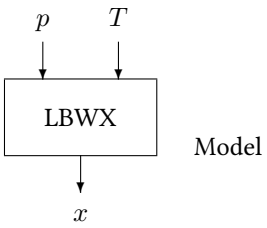
The temperature of lithium-bromide-water dilution is calculated and plotted by a PLOTP block as a function of concentration and pressure.



The pressure and concentration are varied by two nested DO blocks between 1000 and 101000 Pa in steps of 1000 (outer block) and 0 and 0.76 kg LiBr per kg water in steps of 0.01 (inner block).

4.40 Block LBWX

The LBWX block returns the concentration of a lithium-bromide-water dilution as a function of pressure and temperature.



Model

Name	LBWX
Function	st0025
Inputs	2
Outputs	1
Parameters	2
Strings	0
Group	S

Inputs

1 Pressure p / Pa
2 Temperature T / °C

Outputs

1 Lithium-bromide concentration x / kg LiBr kg^{-1} H_2O

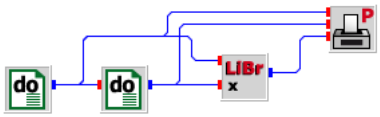
Parameters

1 Model
0 Mc Neely 1979
2 Outside-range value. If set to zero ($|BP(2)| < 10^{-5}$) data are extrapolated.

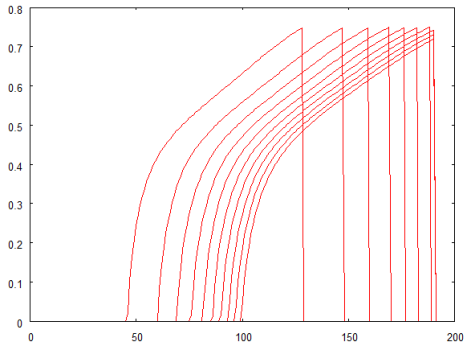
Strings

None

lbwx.vseit



The concentration of lithium-bromide-water dilution is calculated and plotted by a PLOTP block as a function of pressure and temperature.

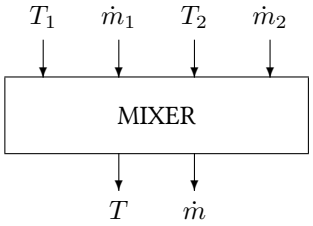


The pressure and temperature are varied by two nested DO blocks between 0 and 200 °C in steps of 1 °C (inner block) and 10000 and 100000 Pascal in steps of 10000 (outer block).

The outside-range parameter is set to 10^{-4} so that the curves end at a maximum concentration of 0.76 kg LiBr per kg water.

4.41 Block MIXER

The MIXER block simulates a mixing valve for two fluids of different temperature and mass flow but with the same specific heat capacity.



Name	MIXER
Function	st0034
Inputs	4
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1	Temperature fluid 1 / °C
2	Mass flow fluid 1 / kg s^{-1}
3	Temperature fluid 2 / °C
4	Mass flow fluid 2 / kg s^{-1}

Outputs

1	Mixed temperature / °C
2	Total mass flow / kg s^{-1}

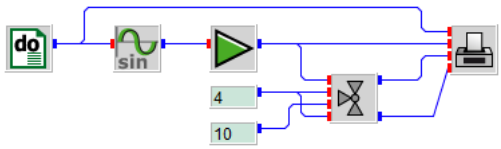
Parameters

None

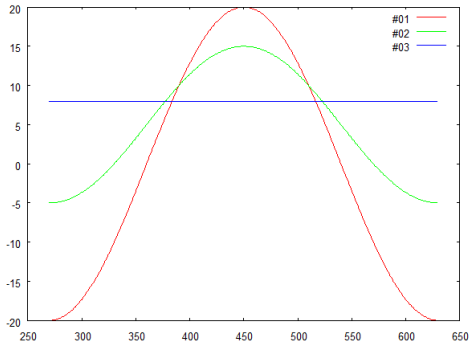
Strings

None

mixer.vseit



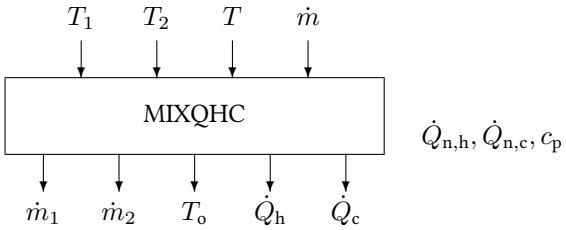
Two fluids at a constant mass flow of 4 kg/s each are mixed by a MIXER block. While the one is held constant at 10 °C the other follows a sine wave between −20 and +20 °C. The result can be observed by the PLOT block.



The total mass flow is 8 kg/s, of course.

4.42 Block MIXQHC

The MIXQHC block simulates a mass flow mixer with auxiliary heating and cooling devices.



Name	MIXQHC
Function	st0011
Inputs	4
Outputs	5
Parameters	3
Strings	0
Group	S

Inputs

1 Temperature fluid one T_1 / °C
2 Temperature fluid two T_2 / °C
3 Mix temperature T / °C
4 Total mass flow $\dot{m}$ / kg s^{-1}

Outputs

1 Mass flow fluid one $\dot{m}_1$ / kg s^{-1}
2 Mass flow fluid two $\dot{m}_2$ / kg s^{-1}
3 Outlet temperature T_o / °C
4 Auxiliary heating power $\dot{Q}_h$ / W
5 Auxiliary cooling power $\dot{Q}_c$ / W

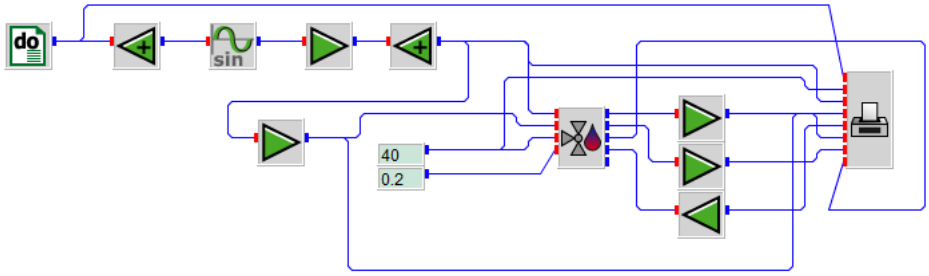
Parameters

1 Nominal power of auxiliary heater $\dot{Q}_{n,h}$ / W
2 Nominal power of auxiliary cooler $\dot{Q}_{n,c}$ / W
3 Specific heat of fluid c_p / J kg^{-1} K^1

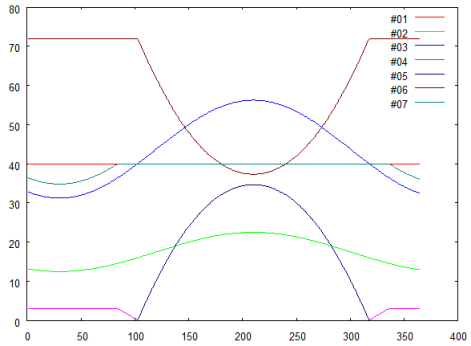
Strings

None

mixqhc.vseit



A DO block with parameters 1, 365, and 1 runs through the days of one year in order to create a test profile for the temperature of a fluid via a SIN block and some gains and offsets. The resulting profile can be seen as green curve number two in the output of the PLOT block.



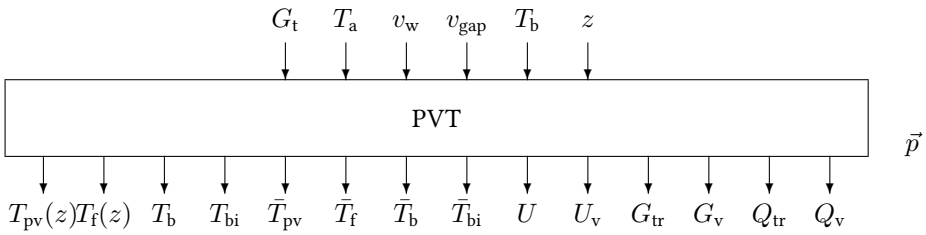
The test signal is connected to the first input of the MIXQHC block. The second fluid temperature follows the same profile but is multiplied by a factor 2.5. The desired mix temperature is set constant to 40 °C and the desired total mass flow is 0.2 kg/s.

The outputs of the MIXQHC block are scaled through two GAIN blocks and an ATT block so that they fit into one diagram.

It can be seen that the required mix temperature of 40 °C cannot be reached in the winter months. Hence, some auxiliary heating power is required (magenta curve number four).

4.43 Block PVT

The PVT block simulates the thermal performance of a building-integrated PV generator within a double facade. The PV module is assumed to face exterior air, followed by a ventilation gap of variable depth and then a back glazing or insulation for the thermal separation to the room.



Name	PVT
Function	st0002
Inputs	6
Outputs	14
Parameters	13
Strings	0
Group	S

Inputs

1 Global radiation on the generator plane G_t / W m^{-2}
2 Ambient temperature T_a / °C
3 Wind speed close to the generator v_w / m s^{-1}
4 Flow velocity behind the generator, i. e., in the gap v_{gap} / m s^{-1}
5 Temperature of generator back surface T_b / °C
6 Height of interest z / m

Outputs

1 PV module temperature $T_{pv}(z)$ / °Cat height z
2 Facade temperature $T_f(z)$ / °Cat height z
3 Temperature of gap surface backside T_b / °C
4 Temperature of gap surface inside T_{bi} / °C
5 Mean PV module temperature $\bar{T}_{pv}$ / °C
6 Mean fluid temperature $\bar{T}_f$ / °C
7 Mean temperature of gap surface backside $\bar{T}_b$ / °C
8 Mean temperature of gap surface inside $\bar{T}_{bi}$ / °C

9 Transmission heat loss coefficient U_{tr} / W m^{-2} K^{-1}
10 Recovered transmission heat loss coefficient U_v / W m^{-2} K^{-1}
11 Total energy transmittance G_{tr}
12 Recovered energy transmittance G_v
13 Transmission heat loss Q_{tr} / kWh m^{-2}
14 Ventilation heat gains Q_v / kWh m^{-2}

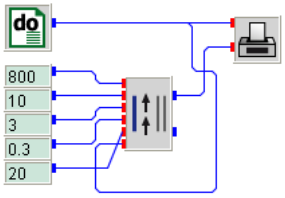
Parameters

1 PV module height / m
2 PV module width / m
3 PV module thickness / m
4 PV module efficiency under standard test conditions
5 PV module emissivity facing exterior air
6 PV module emissivity facing air gap
7 PV absorption coefficient
8 PV module transmission
9 PV module heat conductivity / W m^{-1} K^{-1}
10 Air gap depth / m
11 Back glazing absorption coefficient of facing room
12 Back glazing emissivity facing the air gap
13 Back glazing/insulation heat resistance / m^2 K W^{-1}

Strings

None

pvt.vseit

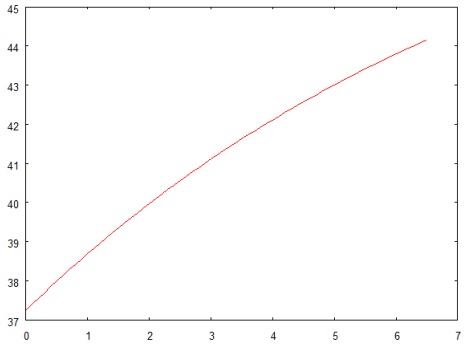


A ventilated PV façade with the following parameters is simulated:

PV module height	6.5
PV module width	0.994
PV module thickness	0.01
PV module efficiency under standard test conditions	0.12
PV module emissivity facing exterior air	0.88
PV module emissivity facing air gap	0.88
PV absorption coefficient	0.8
PV module transmission	0.1
PV module heat conductivity	0.8
Air gap depth	0.14
Back glazing absorption coefficient of facing room	0.05
Back glazing emissivity facing the air gap	0.88
Back glazing/insulation heat resistance	1.5

All inputs of the PVT block are set constant by CONST blocks, except the height, which is varied between zero and the total height of the air gap of 6.5 m. The increment of the DO block is 0.1 m.

The PLOT block displays the PV module temperature as a function of height.



4.44 Block PVTCOOL

This Block models a PV panel combined with a thermal collector. The thermal collector is calculating long-wave radiative heat losses. This is needed for radiative cooling.

Name	PVTCOOL
Function	st0044
Inputs	17
Outputs	12
Parameters	29
Strings	0
Group	S

Inputs

1 eg kJ/(m*m)h globale Bestrahlungsstaerke in der Kollektorebene
2 eb kJ/(m*m)h direkte Bestrahlungsstaerke in der Kollektorebene
3 thetan degree Einfallwinkel der Direktstrahlung
4 thetaz degree Sonnenzenitwinkel
5 gammas degree Sonnenazimutwinkel
6 tu degree celsius Umgebungslufttemperatur
7 pu mbar Umgebungsluftdruck
8 rhu
9 th degree celsius Himmelstemperatur
10 vu m/s Windgeschwindigkeit
11 ukon W/(m*m)K konvektiver Waermeverlustkoeff. des Absorbers / der PV-Modulabdeckung
12 uint W/(m*m)K interner therm. Leitwert Absorber / Fluid
13 tfe degree celsius Arbeitsmitteleintrittstemperatur
14 mpkt kg/h Arbeitsmittelmassenstrom
15 ppvmode - Leistungsabnahme PV-Modul (0:nein / 1: ja)
16 tgr degree celsius Bodentemperatur outputs (Mittelwerte ueber TRNSYS-Zeitschritt)

Outputs

1 mpkt Arbeitsmittelmassenstrom in [kg/h]
2 tfas / nm Arbeitsmittelaustrittstemperatur [degree celsius]
3 mpkt * cf * (out(2) - tfe) / 1000 Nutzwaermestrom in [kJ/h]
4 tams / nm mittlere Absorbertemperatur [celsius]
5 kondensativer waermegewinn in [kJ/h]
6 tgms / nm mittlere Temperatur der PV-Modulabdeckung in [degree celsius]
7 ppvmode * ppvs / nm elektrische Leistung des PV-Moduls [W]
8 ukon konvektiver waermeverlustkoeff. (vorn) des Kollektors [W/(m*m)/K]
9 uint interner thermischer Leitwert Absorber / Fluid [W/(m*m)/K]
10 enx nettobestrahlungsstaerke En* [W/(m*m)]
11 einfallwinkel-korrigierten bestrahlungsstaerke En [W/(m*m)]
12 konvektiven waermeverluste : (b1+b2*u)*(tm-tu) [W/(m*m)]

Parameters

1 kmode [-] Kollektormodus (nur thermisch ohne/mit Kondensationsgewinnen(1/2), photovoltaisch-thermisch ohne/mit Kondensationsgewinnen(3/4),)
2 pvmode [-] PV-Teilmodell (1: Kennlinienmodell / 2: Wirkungsgradmodell)
3 ukmode [-] Berechnungsmodus fuer ukon (1: input / 2: interne Berechnung)
4 uimode [-] Berechnungsmodus fuer uint (1: input / 2: interne Berechnung)
5 aa [m*m] Kollektorflaeche apertur
6 beta [degree] Kollektorneigungswinkel
7 gammaa [degree] Kollektorazimutwinkel
8 eta0 [-] Konversionsfaktor
9 bu [s/m] Windabhaengigkeit des Konversionsfaktors
10 b1 [W/(m*m)K] Waermeverlustkoeffizient
11 b2 [J/m³K] Windabhaengigkeit des Waermeverlustkoeffizienten
12 alphath [-] Absorptionsgrad des Absorbers
13 epsalpha [-] Emissions- / Absorptionsgrad-Verhaeltnis
14 ceff [kJ/(m*m)K] effektive Waermekapazitaet des Kollektors
15 cf [kJ/kgK] spez. Waermekapazitaet des Arbeitsmittels
16 kdth [-] Einfallwinkelkorrekturfaktor fuer Diffusstrahlung, Absorber
17 unrth [-] logical unit no. der Einfallwinkelkorrekturfaktor-Datei, Absorber
18 ugf [W/(m*m)/K] thermischer Leitwert der PV-Modulabdeckung (Glas+Folie)
19 epspv [-] Emissionsgrad der PV-Modulabdeckung
20 taupv [-] Transmissionsgrad der PV-Modulabdeckung
21 kdpv [-] Einfallwinkelkorrekturfaktor fuer Diffusstrahlung, PV-Modul
22 unrpv [-] logical unit no. der Einfallwinkelkorrekturfaktor-Datei, PV-Modul
23 etastc [degree celsius] PV-Modulwirkungsgrad bei Standard-Testbedingungen
24 ik [A] Kurzschlussstrom
25 impp [A] Strom bei Nennleistung
26 ul [V] Leerlaufspannung
27 umpp [V] Spannung bei Nennleistung

28 ct [
29 relax [-] circa 0.6, Relaxationsfaktor des Modellgleichungs-Loesungsalgorithmus
30 epsgr [-] Emissionsgrad des Bodens

Strings

None

4.45 Block R134A

The R134A block calculates the thermodynamic properties of refrigerant R134a.

Name	R134A
Function	st0043
Inputs	2
Outputs	7
Parameters	0
Strings	0
Group	S

Inputs

1 Temperature / °C
2 Density / $\mathrm{kg\,m^{-3}}$

Parameters

None

Outputs

1 Pressure p / MPa
2 Specific enthalpy h / $\mathrm{kJ\,kg^{-1}}$
3 Specific entropy s / $\mathrm{kJ\,kg^{-1}\,K^{-1}}$
4 Specific isobaric heat capacity / $\mathrm{kJ\,kg^{-1}\,K^{-1}}$
5 Vapor pressure p_s / MPa
6 Densitiy ρ'/ $\mathrm{kg\,m^{-3}}$
7 Specific enthalpy h' / $\mathrm{kJ\,kg^{-1}}$

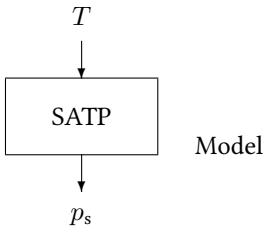
Strings

None

4.46 Block SATP

The SATP block returns water vapour saturation pressure as a function of temperature.

SATP



Name	SATP
Function	st0009
Inputs	1
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Temperature T / °C

Outputs

1 Water vapour saturation pressure p_s / Pa

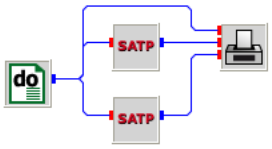
Parameters

1 Mode
- 0 Y. O. Devres
  Psychrometric properties of humid air: Calculation procedures
  in: Applied Energy 48 1994 (accuracy: ± 5 %)
- 1 W. Wagner and A. Pruss

Strings

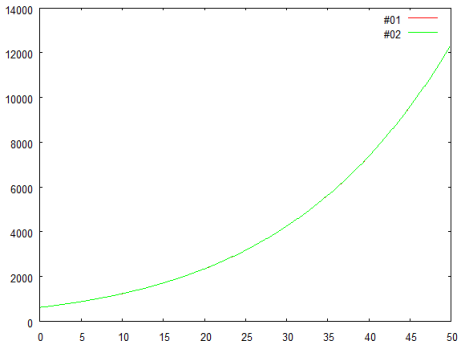
None

satp.vseit



The saturation pressure of water vapor is calculated according to approximations of Devres (upper block) and Wagner Pruss (lower block) for dew point temperatures which are varied between 0 and 50 °C by a DO block in steps of 0.1 °C.

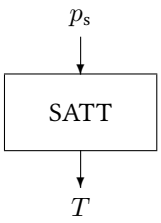
The PLOT block shows that the difference between the two approximations is very small.



See also Block SATT.

4.47 Block SATT

The SATT block returns water vapour saturation temperature (dew point temperature) as a function of pressure.



Model

Name	SATT
Function	st0009
Inputs	1
Outputs	1
Parameters	0 . . . [1]
Strings	0
Group	S

Inputs

1 Water vapour saturation pressure p_s / Pa

Outputs

1 Dew point temperature T / °C

Parameters

1 Mode

0 Y. O. Devres
Psychrometric properties of humid air: Calculation procedures
in: Applied Energy 48 1994 (accuracy: ± 5 %)

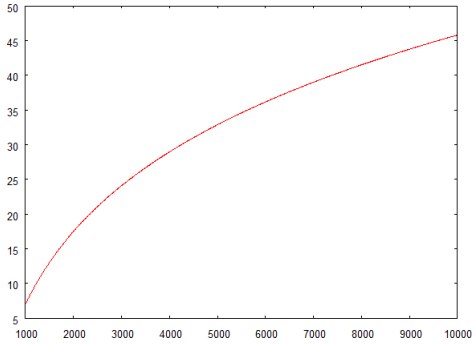
Strings

None

satt.vseit

The saturation temperature of water vapor is calculated after an approximation by Devres. The required water vapor saturation pressure is varied between 1000 and 10 000 Pa in steps of 10 Pa by a DO block.

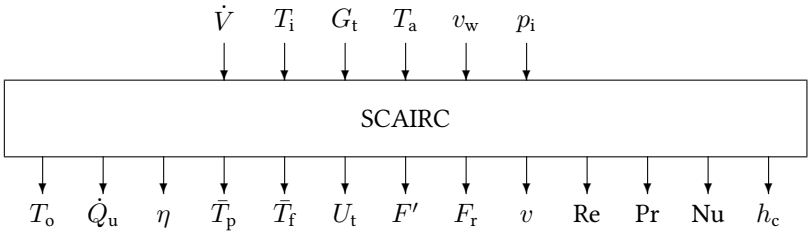
A PLOT block shows the result.



See also Block SATP.

4.48 Block SCAIRC

The SCAIRC block simulates a solar air collector with a ripped absorber (channel structure).



Name	SCAIRC
Function	st0001
Inputs	5 . . . [6]
Outputs	13
Parameters	18
Strings	0
Group	S

Inputs

1 Volume flow $\dot{V}$ / $m_n^3\,s^{-1}$
2 Inlet temperature T_i / °C
3 Global radiation on the collector plane G_t / $W\,m^{-2}$
4 Ambient temperature T_a / °C
5 Wind speed close to the collector v_w / $m\,s^{-1}$
6 Inlet pressure p_i / Pa (optional)

Outputs

1 Outlet temperatur T_o / °C
2 Useful heat $\dot{Q}_u$ / W
3 Efficiency η
4 Mean absorber plate temperature $\bar{T}_p$ / °C
5 Mean fluid temperature $\bar{T}_f$ / °C
6 Overall heat loss coefficient U_t / $W\,m^{-2}\,K^{-1}$
7 Absorber efficiency factor F'
8 Heat removal factor F_r
9 Flow speed v / $m\,s^{-1}$
10 Reynolds number Re

11	Prandtl number Pr
12	Nusselt number Nu
13	Convective heat transfer coefficient h_c / W m^{-2} K^{-1}

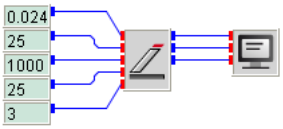
Parameters

1	Number of collectors in series N_s
2	Number of collectors in parallel N_p
3	Collector tilt angle β / °
4	Collector length l_c / m
5	Collector width w_c / m
6	Channel height h_{ch} / m
7	Channel width w_{ch} / m
8	Number of channels N_{ch}
9	Plate thickness t_p / m
10	Optical efficiency $(\tau\alpha)$
11	Thickness of insulation material t_i / m
12	Heat conductivity of insulation material λ_i / W m^{-1} K^{-1}
13	Heat conductivity of absorber material λ_a / W m^{-1} K^{-1}
14	Emissivity of glas cover ϵ_c
15	Emissivity of absorber plate front side towards cover ϵ_f
16	Emissivity of absorber plate back side towards channel ϵ_b
17	Emissivity of channel towards absorber plate back side ϵ_{ch}
18	Distance between absorber plate and front cover d / m

Strings

None

scairc.vseit



A solar air collector is simulated with the following parameters.

```
Number of collectors in series                          1
Number of collectors in parallel                        1
Collector tilt angle                                   45
Collector length                                        2.5
Collector width                                         0.96
Channel height                                          0.028
Channel width                                           0.026
Number of channels                                     36
Plate thickness                                         0.0006
Optical efficiency                                      0.812
Thickness of insulation material                        0.06
Heat conductivity of insulation material                0.04
Heat conductivity of absorber material                238
Emissivity of glas cover                                0.88
Emissivity of absorber plate front side towards cover   0.89
Emissivity of absorber plate back side towards channel  0.085
Emissivity of channel towards absorber plate back side  0.085
Distance between absorber plate and front cover         0.020
```

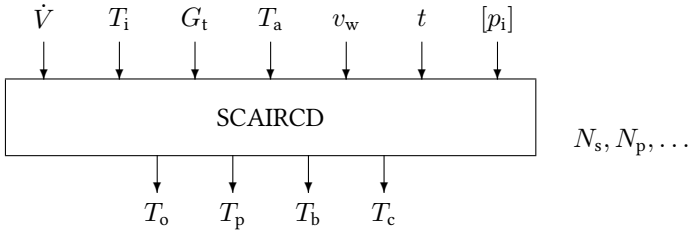
All inputs are set constant by CONST blocks. A SCREEN block displays outlet temperature, useful heat and the collector efficiency.

```
   65.62508  1069.95264     0.44581
```

See also Block SCAIRCD.

4.49 Block SCAIRCD

The SCAIRCD block simulates a solar air collector with a ripped absorber (channel structure) under consideration of the thermal heat capcity.



Name	SCAIRCD
Function	st0029
Inputs	6 . . . [7]
Outputs	4
Parameters	23
Strings	0
Group	S

Inputs

1 Volume flow $\dot{V}$ / $\mathrm{m^3\,s^{-1}}$
2 Inlet temperature T_i / °C
3 Global radiation on the collector plane G_t / $\mathrm{W\,m^{-2}}$
4 Ambient temperature T_a / °C
5 Wind speed close to the collector v_w / $\mathrm{m\,s^{-1}}$
6 Time t / s
7 Inlet pressure p_i / Pa (optional)

Outputs

1 Outlet air temperature T_o / °C
2 Temperature absorber plate at outlet T_p / °C
3 Temperature back side at outlet T_b / °C
4 Temperature front cover at outlet T_c / °C

Parameters

1 Number of collectors in series N_s
2 Number of collectors in parallel N_p
3 Collector tilt angle β / °

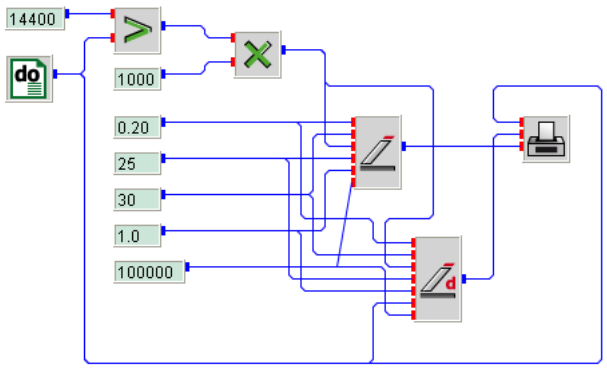
4 Collector length l_c / m
5 Collector width w_c / m
6 Channel height h_{ch} / m
7 Channel width w_{ch} / m
8 Number of channels N_{ch}
9 Plate thickness t_p / m
10 Optical efficiency $(\tau\alpha)$
11 Thickness of insulation material t_i / m
12 Heat conductivity of insulation material λ_i / $\mathrm{W\,m^{-1}\,K^{-1}}$
13 Heat conductivity of absorber material λ_a / $\mathrm{W\,m^{-1}\,K^{-1}}$
14 Emissivity of glas cover ϵ_c
15 Emissivity of absorber plate front side towards cover ϵ_f
16 Emissivity of absorber plate back side towards channel ϵ_b
17 Emissivity of channel towards absorber plate back side ϵ_{ch}
18 Distance between absorber plate and front cover d / m
19 Thickness of back cover material d_b / m
20 Heat capacity of back cover material $c_{p,b}$ / $\mathrm{J\,kg^{-1}\,K^{-1}}$
21 Density of back cover material ρ_b / $\mathrm{kg\,m^{-3}}$
22 Heat capacity of absorber material $c_{p,a}$ / $\mathrm{J\,kg^{-1}\,K^{-1}}$
23 Density of absorber material ρ_a / $\mathrm{kg\,m^{-3}}$

Strings

None

---

scaircd.vseit



In this example the static model used by the SCAIRC block is compared to the dynamic model, implemented in block SCAIRCD.

The parameters of the static SCAIRC block are set to these values:

```
Number of collectors in series                          2
Number of collectors in parallel                        2
Collector tilt angle                                   34
Collector length                                     12.5
Collector width                                      0.96
Channel height                                      0.095
Channel width                                       0.060
Number of channels                                     16
Plate thickness                                     0.0014
Optical efficiency                                     0.8
Thickness of insulation material                      0.06
Heat conductivity of insulation material              0.04
Heat conductivity of absorber material                 238
Emissivity of glas cover                              0.88
Emissivity of absorber plate front side towards cover 0.16
Emissivity of absorber plate back side towards channel 0.040
Emissivity of channel towards absorber plate back side 0.040
Distance between absorber plate and front cover       0.025
```

The dynamic SCAIRCD block uses these parameters.

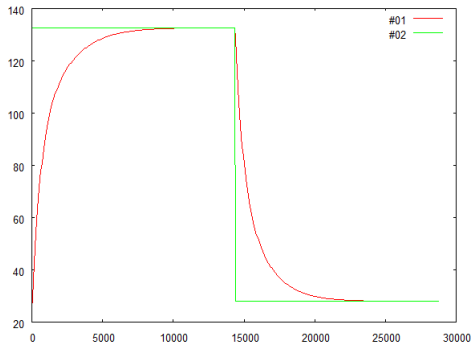
```
Number of collectors in series                          2
Number of collectors in parallel                        2
Collector tilt angle                                   34
Collector length                                     12.5
Collector width                                      0.96
Channel height                                      0.095
Channel width                                       0.060
Number of channels                                     16
Plate thickness                                     0.0014
Optical efficiency                                   0.812
Thickness of insulation material                      0.06
Heat conductivity of insulation material              0.04
Heat conductivity of absorber material                 238
Emissivity of glas cover                              0.88
Emissivity of absorber plate front side towards cover 0.16
Emissivity of absorber plate back side towards channel 0.085
Emissivity of channel towards absorber plate back side 0.085
Distance between absorber plate and front cover       0.020
Thickness of back cover material                      0.002
Heat capacity of back cover material                  500
Density of back cover material                       7800
Heat capacity of absorber material                    500
Density of absorber material                         2702
```

Most of the inputs of the two air collector blocks are set constant by CONST blocks.

However, the time is varied between 0 and 28800 seconds (eight hours) in steps of 60 seconds by a DO block.

The collector irradiance of 1000 W/m^2, as defined by one of the CONST blocks is switched off after 14400 seconds. This is accomplished by a CONST block with parameter 14400 and a GT block. When the time signal, coming from the DO block gets greater than 14400, the GT block outputs zero and the MUL block "switches the irradiance off".

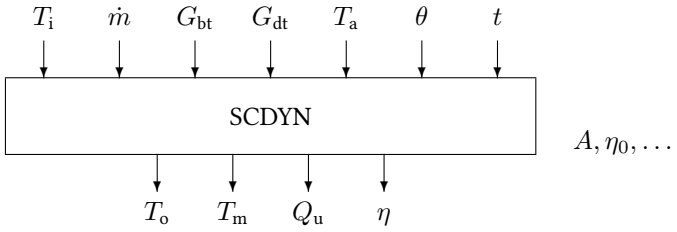
The PLOT block demonstrates the result.



See also Block SCAIRC.

4.50 Block SCDYN

The SCDYN block simulates a solar vacuum tube and CPC collector under dynamic conditions according to the European Standard EN 12975. This model is a 1 node model that allows the partition of the collector in differents elements in order to increase the accuracy the model and avoid the peak in the outlet temperature caused by the switching on of the pump after a time of stopping of the pump.



Name	SCDYN
Function	st0027
Inputs	7
Outputs	4
Parameters	11
Strings	0
Group	S

Inputs

1 Inlet temperature T_i / °C
2 Mass flow rate $\dot{m}$ / kg s^{-1}
3 Beam radiation in the collector plane G_{bt} / W m^{-2}
4 Diffuse radiation in the collector plane G_{dt} / W m^{-2}
5 Ambient temperature T_a / °C
6 Incidence angle of beam radiation θ / degrees
7 Time t / s

Outputs

1 Outlet temperature of the collector T_o/°C
2 Mean temperature of the collector T_m/°C
3 Useful heat of the collector $\dot{Q}_u$ / W
4 Global collector efficiency η

Parameters

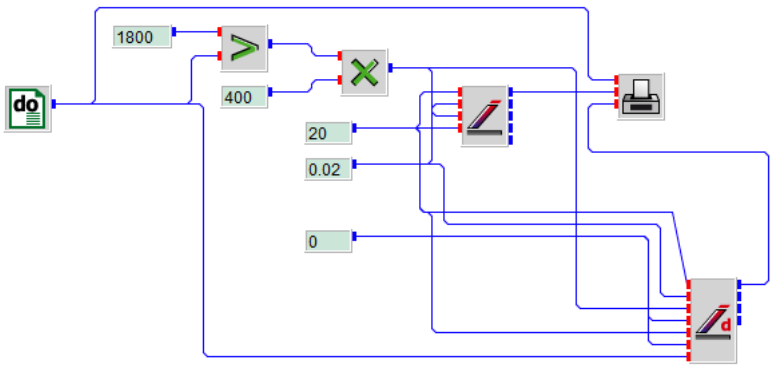
1 Collector area / m^2
2 Zero loss efficiency at normal incidence η_0
3 Linear heat loss coefficient a_1 / $W\,m^{-2}\,K^{-1}$
4 Quadratic heat loss coefficient a_1 / $W\,m^{-2}\,K^{-2}$
5 Incidence angle modifier (IAM) for diffuse radiation K_d
6 Effective thermal capacity of the collector c_{eff} / $J\,m^{-2}\,K^{-1}$
7 Heat capacity of the fluid c_p $J\,kg^{-1}\,K^{-1}$
8 Number of collectors in series N_s
9 Number of collectors in parallel N_p
10 Initial mean temperature of the collector $T_{m,0}/°C$
11 Number of partition elements per collector N_{seg}

Strings

None

---

scdyn.vseit



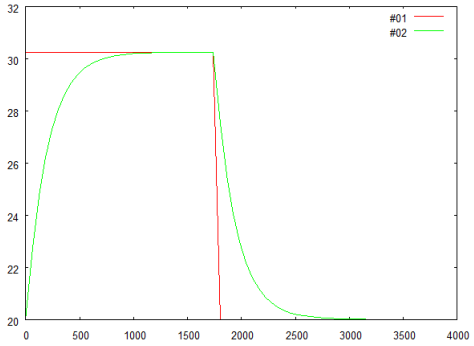
In this example the static model used by the SCETA block is compared to the dynamic model, implemented in block SCDYN. In both cases a single collector with an absorber area of 3 m^2, optical efficiency of 0.675, linear heat loss coefficient 0.697 $W\,m^{-2}\,K^{-1}$ and quadratic heat loss coefficient 0.003 $W\,m^{-2}\,K^{-2}$ is simulated.

Most of the inputs of the two collector blocks are set constant by CONST blocks.

However, the time is varied between 0 and 3600 seconds (one hour) in steps of 60 seconds by a DO block.

The collector irradiance of 400 W/m^2, as defined by one of the CONST blocks is switched off after 1800 seconds. This is accomplished by a CONST block with parameter 1800 and a GT block. When the time signal, coming from the DO block gets greater than 1800, the GT block outputs zero and the MUL block "switches the irradiance off".

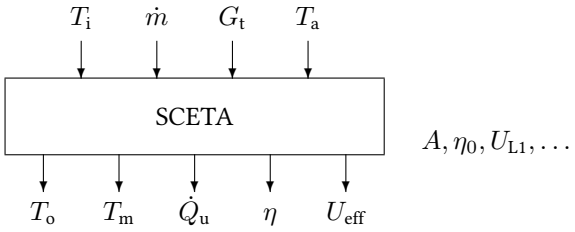
The PLOT block demonstrates the result.



See also Block SCAIRC.

4.51 Block SCETA

The SCETA block simulates a solar collector on the basis of the collector equation.



Name	SCETA
Function	st0008
Inputs	4
Outputs	5
Parameters	7 . . . [8]
Strings	0 . . . [1]
Group	S

Inputs

1 Inlet temperature T_i / °C
2 Mass flow rate $\dot{m}$ / kg s^{-1} (or mean fluid temperature T_m / °C: mode 1)
3 Global radiation on the generator plane G_t / W m^{-2}
4 Ambient temperature T_a / °C

Outputs

1 Outlet temperature T_o / °C
2 Mean fluid temperature T_m / °C
3 Useful heat $\dot{Q}_u$ / W
4 Efficiency η (can be greater than one due to convective gains)
5 Effective heat-loss coefficient U_{eff} / W m^{-2} K^{-1}

Parameters

1 Absorber area A / m^2
2 Maximum efficiency $\eta_0 = (\tau\alpha)F'$
3 Efficiency parameter U_{L1} / W m^{-2} K^{-1}
4 Efficiency parameter U_{L2} / W m^{-2} K^{-2}
5 Specific heat of collector fluid c_p / J kg^{-1} K^1
6 Number of collectors in series N_s

7 Number of collectors in parallel N_p
8 Temperature mode
0 IN(2) = Mass flow rate (default)
1 IN(2) = Mean fluid temperature

Strings

1 Product ID

Description

The collector equation is given by

$$\eta = \eta_0 - U_{L1}\frac{T_\mathrm{m} - T_\mathrm{a}}{G} - U_{L2}\frac{(T_\mathrm{m} - T_\mathrm{a})^2}{G}$$

with the mean fluid temperature T_m, G being the global radiation in the collector plane and the three fit parameters $\eta_0, U_{\mathrm{L}1}$, and $U_{\mathrm{L}2}$. The mean fluid temperature T_m may be approximated by the mean value of the collector inlet temperature T_i and the collector outlet temperature T_o

$$T_\mathrm{m} = \frac{T_\mathrm{i} + T_\mathrm{o}}{2}$$

The useful power is then given by

$$\dot{Q}_\mathrm{u} = \eta G A = \dot{m} c_\mathrm{p} (T_\mathrm{o} - T_\mathrm{i})$$

where A represents the absorber area and $\dot{m}$ the mass flow rate.

In case $U_{L2} = 0$ the outlet temperature is found to be

$$T_\mathrm{o} = \frac{\eta_0 G + T_\mathrm{i}\dot{m}c_\mathrm{p}/A - U_{\mathrm{L}1}T_\mathrm{i}/2 + U_{\mathrm{L}1}T_\mathrm{a}}{\dot{m}c_\mathrm{p}/A + U_{\mathrm{L}1}/2}$$

T_o cannot exceed the stagnation temperature

$$T_\mathrm{max} = \frac{1}{U_{\mathrm{L}1}}\left(\eta_0 G - \frac{U_{\mathrm{L}1}T_\mathrm{i}}{2} + U_{\mathrm{L}1}T_\mathrm{a}\right)$$

The case where $U_{L2} \neq 0$ leads to a slightly more complex quadratic equation with the solutions

$$T_{\mathrm{o}1/2} = -\frac{p}{2} \pm \sqrt{(\frac{p}{2})^2 - q}$$

with the coefficients

$$p = \frac{2U_{\mathrm{L}1}}{U_{\mathrm{L}2}} + 2T_\mathrm{i} - 4T_\mathrm{a} + \frac{2\dot{m}c_\mathrm{p}}{AU_{\mathrm{L}2}}$$

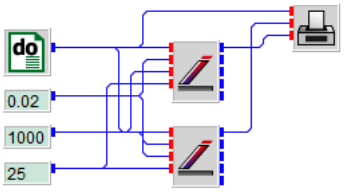
and

$$q = \left(2\frac{U_{L1}}{U_{L2}} - 4T_a + T_i - \frac{4\dot{m}c_p}{AU_{L2}}\right)T_i - 4\left(\frac{G\eta_0}{U_{L2}} + \frac{U_{L1}T_a}{U_{L2}} + {T_a}^2\right)$$

The effective heat-loss coefficient U_{eff} is calculated from

$$U_{eff} = U_{L1} + U_{L2}(T_m - T_a)$$

sceta.vseit

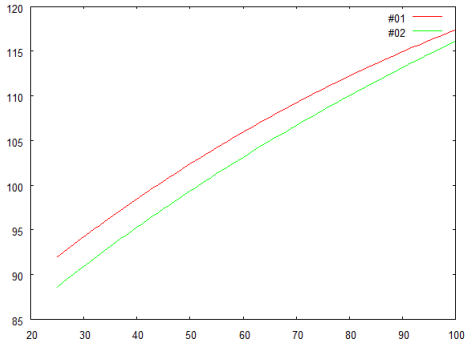


Two SCETA blocks are used to compare the influence of serial and parallel connections of solar collectors. The upper SCETA block simulates eight collectors connected in series, the lower one simulates eight collectors in parallel. Both use a specific heat of the collector fluid of 3900 J kg^{-1} K^{-1} (water) and the temperature mode "IN(2) = Mass flow rate".

The mass flow is set to 0.02 kg/s, the irradiance to 1000 W/m^2, and the ambient temperature to 25 °C by three CONST blocks.

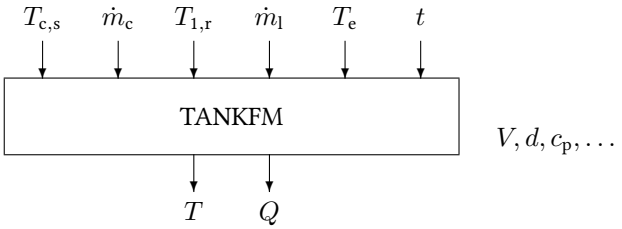
A DO block varies the collector inlet temperature from 25 to 100 °C in steps of 1 °C.

The PLOT block displays the two different collector outlet temperatures.



4.52 Block TANKFM

The TANKFM block simulates a fully-mixed storage tank.



Name	TANKFM
Function	st0028
Inputs	6
Outputs	2
Parameters	6
Strings	0
Group	S

Inputs

1 Supply temperature from collector $T_{c,s}$ / °C
2 Collector mass flow rate $\dot{m}_c$ / kg s^{-1}
3 Return temperature from load $T_{l,r}$ / °C
4 Load mass flow rate $\dot{m}_l$ / kg s^{-1}
5 Environment temperature T_e / °C
6 Time / s

Outputs

1 Storage temperature T / °C
2 Energy content of the storage Q / J

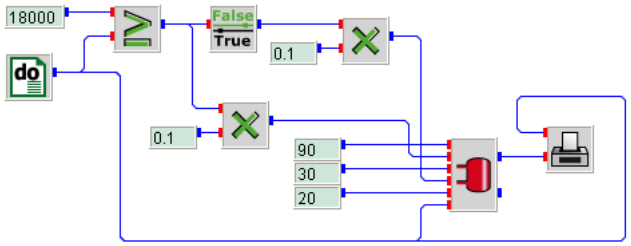
Parameters

1 Tank volume V / m^3
2 Tank diameter d / m
3 Specfic heat of fluid c_p / J kg^{-1} K^{-1}
4 Fluid density ρ / kg m^{-3}
5 Overall heat-loss coefficient U / W m^{-2} K^{-1}
6 Initial tank temperature T_0 / °C

Strings

None

tankfm.vseit

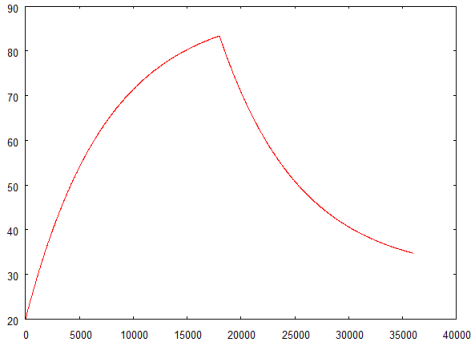


A fully mixed storage tank with a volume of 0.75 m^3 and a diameter of 0.7 m is simulated over a time period of 36 000 seconds (10 hours) in steps of 20 seconds, as defined by a DO block. The tank fluid is assumed to be water with a specific heat capacity of 4190 $J\,kg^{-1}\,K^{-1}$ and a density of 1000 kg/m^3. The overall heat-loss coefficient is set to 0.5 $W\,m^{-2}\,K^{-1}$ at an initial tank temperature of 20 °C.

The supply temperature of the tank is set constant to 90 °C, the return temperature is assumed to be 30 °C, and the environment temperature of the tank is set to 20 °C, as defined by three CONST blocks.

The logics implies that the tank is charged for 18000 seconds (5 hours) at a mass flow of of 0.1 kg/s and then discharged with the same mass flow. As long as the value 18 000, defined by a CONST block is greater or equal to the output of the DO block, the GE block outputs one. This signal, multiplied by the constant mass flow rate of 0.1 kg/s by a MUL block gives the mass flow rate at 90 °C into the tank. The inverse signal of the hot water supply mass flow, generated by the INV block, is used as load mass flow of the tank.

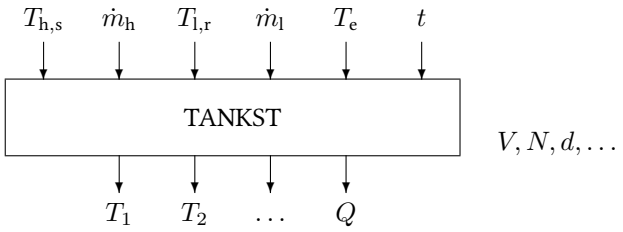
The PLOT block shows the resulting temperature profile in the tank.



See also Block TANKST.

4.53 Block TANKST

The TANKST block simulates a stratified storage tank which can be used either as hot-water storage tank (case 1) or as cold-water storage tank (case 2).



Name	TANKST
Function	st0003
Inputs	6
Outputs	$N + 1$
Parameters	8
Strings	0
Group	D

Inputs

1	Supply temperature heat source $T_{h,s}$/ °C: Case 1 (Hot-water storage tank): From collector/boiler Case 2 (Cold-water storage tank): From load
2	Heat source mass flow rate $\dot{m}_h$ / kg s^{-1}
3	Return temperature cold source $T_{l,r}$/ °C: Case 1 (Hot-water storage tank): From load Case 2 (Cold-water storage tank): From chiller
4	Cold source mass flow rate $\dot{m}_l$ / kg s^{-1}
5	Environment temperature T_e / °C
6	Time / s

Outputs

1	Temperature of node 1 (top) T_1 / °C
2	Temperature of node 2 T_2 / °C
N	Temperature of node N (bottom) T_N / °C
N+1	Energy content of the storage Q / J

Parameters

1	Tank volume V / m^3
2	Number of temperature nodes $N \leq 200$

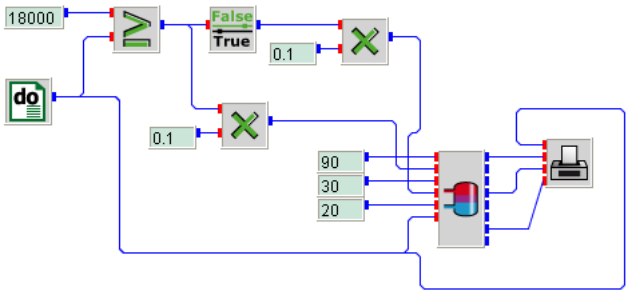
3 Tank diameter d / m
4 Specfic heat of fluid c_p / J kg^{-1} K^{-1}
5 Fluid density ρ / kg m^{-3}
6 Overall heat-loss coefficient U / W m^{-2} K^{-1}
7 Effective heat conductivity λ_{eff} / W m^{-1} K^{-1}
8 Initial tank temperature T_0 / °C

Strings

None

---

tankst.vseit



A stratified storage tank with a volume of 0.75 m^3 and a diameter of 0.7 m is simulated over a time period of 36 000 seconds (10 hours) in steps of 20 seconds, as defined by a DO block. The tank fluid is assumed to be water with a specific heat capacity of 4190 J kg^{-1} K^{-1} and a density of 1000 kg/m^3. The overall heat-loss coefficient is set to 0.5 W m^{-2} K^{-1} at an initial tank temperature of 20 °C.

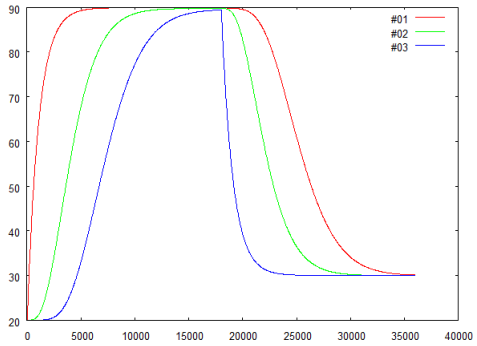
The number of temperature nodes is set to 7, the effective heat conductivity is assumed to be 1 W m^{-1} K^{-1}.

The supply temperature of the tank is set constant to 90 °C, the return temperature is assumed to be 30 °C, and the environment temperature of the tank is set to 20 °C, as defined by three CONST blocks.

The logics implies that the tank is charged for 18000 seconds (5 hours) at a mass flow of of 0.1 kg/s and then discharged with the same mass flow. As long as the value 18 000, defined by a CONST block is greater or equal to the output of the DO block, the GE block outputs one. This signal, multiplied by the constant mass flow rate of 0.1 kg/s by a MUL block gives the mass flow rate at 90 °C into the tank.

The inverse signal of the hot water supply mass flow, generated by the INV block, is used as load mass flow of the tank.

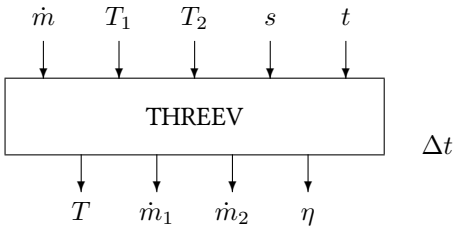
The PLOT block shows the resulting temperature profile in the tank for nodes 1 (top), node 4 (middle), and seven (bottom).



See also Block TANKFM.

4.54 Block THREEV

The THREEV block simulates a three way mixing valve including the valve dynamics.



Name	THREEV
Function	st0041
Inputs	5
Outputs	4
Parameters	1
Strings	0
Group	S

Inputs

1	Total mass flow rate / kg s^{-1}
2	Temperature of source 1 / °C
3	Temperature of source 2 / °C
4	Control signal (0–100 %) (1...-1)
5	Time / s

Outputs

1	Mixed temperature / °C
2	Mass flow medium 1 / kg s^{-1}
3	Mass flow medium 2 / kg s^{-1}
4	Valve position / %

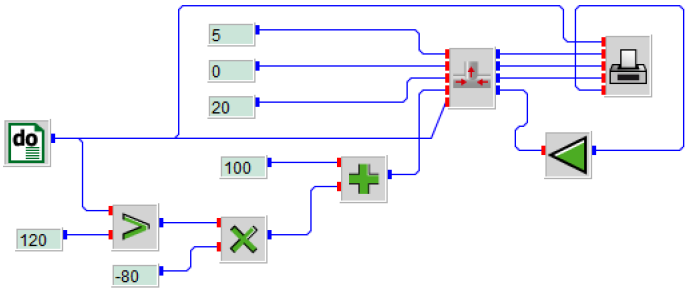
Parameters

1	Time to fully open valve (0–100%) / s

Strings

None

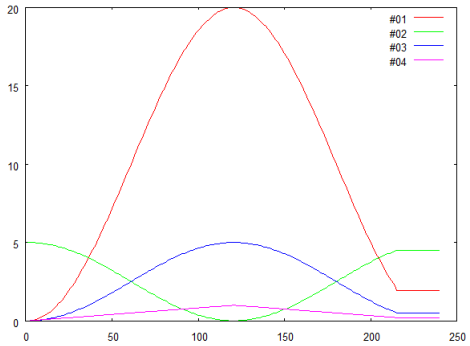
threev.vseit



A three-way valve is supplied with media at temperatures of zero and twenty degrees. A total massflow of 5 kg/s shall be achieved. The dynamic opening time of the valve is set to 120 seconds, i. e., two minutes.

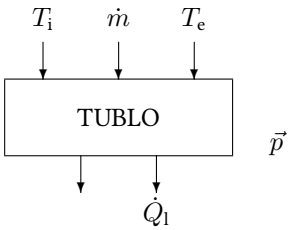
A DO block provides the time (240 seconds). At the beginning the valve gets a control signal of 100 percent. After two minutes this signal is restricted to 20 percent through the logics of the GT block.

A PLOT block is used to show what happens. The magenta curve shows the normalized control signal. The red signal is the achieved mix temperature.



4.55 Block TUBLO

The TUBLO block calculates the heat losses through tubes using the radial heat loss coefficient and the total tube length.



Name	TUBLO
Function	st0042
Inputs	3
Outputs	2
Parameters	4
Strings	0
Group	S

Inputs

1 Tube inlet temperature / °C
2 Fluid mass flow / kg s^{-1}
3 Adjacent air temperature / °C

Parameters

1 Radial heat loss coefficient of the isolated tube / W m^{-2} K^{-1}
2 Increase of heat losses through ageing and imperfect isolation / %
3 Heat capacity of the fluid / kJ kg^{-1} K^{-1}
4 Total length of tubes / m

Outputs

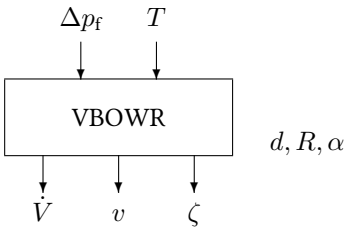
1 Tube outlet temperature / °C
2 Heat losses of the tube / kW

Strings

None

4.56 Block VBOWR

The VBOWR block calculates the Volume flow of a round bow due to friction of an incompressible one-phase turbulent tubular flow.



Name	VBOWR
Function	st0021
Inputs	2
Outputs	3
Parameters	3
Strings	0
Group	S

Inputs

1 Pressure drop Δp_f / Pa
2 Air temperature T / °C

Outputs

1 Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2 Flow velocity v / $m\,s^{-1}$
3 Zeta value ζ

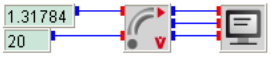
Parameters

1 Tube diameter d / m
2 Bow radius R / m
3 Angle α / °

Strings

None

vbowr.vseit



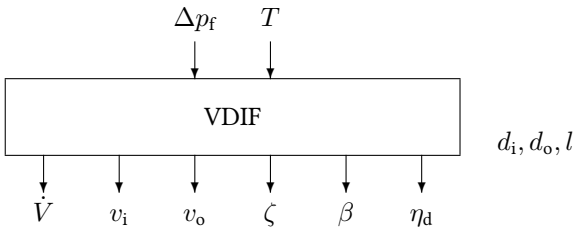
A round 90° bow with a tube diameter of 0.4 m and a bow radius of 1 m is simulated for a pressure drop of 1.31784 Pa and an air temperature of 20 °C as defined by two CONST blocks.

A SCREEN block displays the three outputs of the VBOWR block.

```
1800.00269     3.97888     0.14012
```

4.57 Block VDIF

The VDIF block calculates the Volume flow in a diffuser due to friction of an incompressible one-phase turbulent tubular flow.



Name	VDIF
Function	st0018
Inputs	2
Outputs	6
Parameters	3
Strings	0
Group	S

Inputs

1 Pressure drop Δp_f / Pa
2 Air temperature T / °C

Outputs

1 Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2 Inlet flow velocity v_i / $m\,s^{-1}$
3 Outlet flow velocity v_o / $m\,s^{-1}$
4 Zeta value ζ
5 Opening angle β / $m\,s^{-1}$
6 Diffuser efficiency η_d

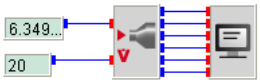
Parameters

1 Inlet diameter d_i / m
2 Outlet diameter d_o / m
3 Length l / m

Strings

None

vdif.vseit



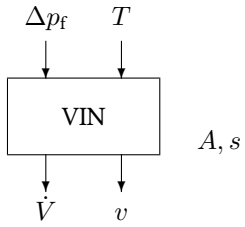
A diffuser of length 1 m with an inlet diamter of 0.25 m and an outlet diamter of 0.4 m is simulated for a pressure drop of 6.34972 Pa and an air temperature of 20 °C as defined by two CONST blocks.

A SCREEN block displays all six outputs of the VDIF block.

```
1800.00049  10.18592   3.97887   0.10302   8.57831   0.87843
```

4.58 Block VIN

The VIN block calculates the volume flow from free space into a channel or tube.



Name	VIN
Function	st0019
Inputs	2
Outputs	2
Parameters	2
Strings	0
Group	S

Inputs

1 Pressure drop Δp_f / Pa
2 Air temperature T / °C

Outputs

1 Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2 Flow velocity v / $m\,s^{-1}$

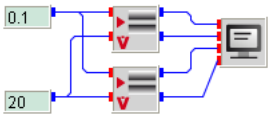
Parameters

1 Channel area A / m^2
2 Edge sharpness s
- 0 Sharp edge
- 1 Broken edge
- 2 Round edge
- 3 Smooth, round edge

Strings

None

vin.vseit



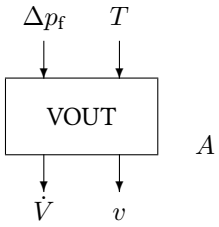
Two different channel inlets with an area of 1 m^2 each, the upper one with a sharp edge, the lower one with a broken edge are simulated for a pressure drop of 0.1 Pa and an air temperature of 20 °C, as defined by two CONST blocks.

The resulting volue flows and flow velocities are displayed by a SCREEN block.

```
2496.5774       0.69349372       4453.3076       1.2370299
```

4.59 Block VOUT

The VOUT block calculates the volume flow for a channel or tube outlet into free space.



Name	VOUT
Function	st0019
Inputs	2
Outputs	2
Parameters	1
Strings	0
Group	S

Inputs

1 Pressure drop Δp_f / Pa
2 Air temperature T / °C

Outputs

1 Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2 Flow velocity v / $m\,s^{-1}$

Parameters

1 Channel area A / m^2

Strings

None

vout.vseit

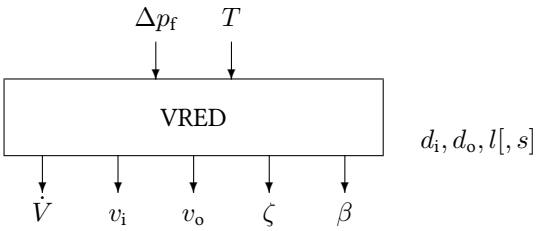
An outlet of 1 m^2 is simulated for a pressure drop of 1 Pa and an air temperature of 20 °C, as defined by two CONST blocks.

The resulting volue flow and flow velocity are displayed by a SCREEN block.

```
4670.6685     1.2974
```

4.60 Block VRED

The VRED block calculates the volume flow in a reducer due to friction of an incompressible one-phase turbulent tubular flow.



Name	VRED
Function	st0020
Inputs	2
Outputs	5
Parameters	3 . . . [4]
Strings	0
Group	S

Inputs

1 Pressure drop Δp_f / Pa
2 Air temperature T / °C

Outputs

1 Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2 Inlet flow velocity v_i / $m\,s^{-1}$
3 Outlet flow velocity v_o / $m\,s^{-1}$
4 Zeta value ζ
5 Closing angle β / $m\,s^{-1}$

Parameters

1 Inlet diameter d_i / m
2 Outlet diameter d_o / m
3 Length l / m
4 Sharpness s (evaluated only if BP(3) = 0)
  0 Sharp edge
  1 Broken edge
  2 Round edge

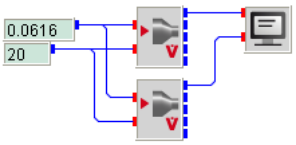
3 Smooth, round edge

Strings

None

---

vred.vseit



Two VRED blocks with different edge sharpnesses are comapred for a pressure drop of 0.0616 Pa and an air temperature of 20 °C as defined by two CONST blocks.

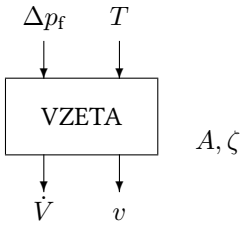
The inlet diameter is set to 0.4, the outlet diameter to 0.25 m, respectively. The length of the reducer is assumed to be zero, i. e., a 90 ° reducer is simulated.

The SCREEN block displays the resulting volume flows.

```
103.08892 162.99789
```

4.61 Block VZETA

The VZETA block calculates the volume flow for a given zeta value.



Name	VZETA
Function	st0019
Inputs	2
Outputs	2
Parameters	2
Strings	0
Group	S

Inputs

1 Pressure drop Δp_f / Pa
2 Air temperature T / °C

Outputs

1 Volume flow $\dot{V}$ / $m^3\,h^{-1}$
2 Flow velocity v / $m\,s^{-1}$

Parameters

1 Channel area A / m^2
2 Zeta value ζ

Strings

None

vzeta.vseit

Assuming a pressure drop of 1 Pascal and an air temperature of 20 °C as defined by two CONST blocks the VZETA block calculares the volume flow through a 1 m^2 channel with a zeta value of 0.3.

The SCREEN block displays the resulting volume flow and flow velocity.

```
8527.4  2.36873
```

5 :: Building Simulation

5.1 Block AMBIENT

The AMBIENT block calculates the convective heat transfer to wall surfaces using the surface temperatures from the wall as an input and calculating heat transfer coefficients from the wind velocity. The block also calculates the longwave radiative heat transfer between the wall surface and the sky and neighbouring buildings. The outputs of the AMBIENT block are the convective and radiative heat flux to each surface temperature node.

Name	AMBIENT
Function	bs0005
Inputs	2 + nwalls
Outputs	2*nwalls
Parameters	0
Strings	0
Group	S

Inputs

1 Ambient temperature (°C)
2 Wind velocity (m/s)
3 Surface temperature tsurf wall i

Outputs

1 Specific convective heat flux to surface node 1 (W/m^2)
2 Specific radiative heat flux to surface node 1 (W/m^2)
2i-1 Specific convective heat flux to surface node i (W/m^2)
. . . OUT(2*nwalls-1) Specific convective heat flux to surface node i (W/m^2)
2i Specific radiative heat flux to surface node i (W/m^2)
. . .
2*nwalls Specific radiative heat flux to surface node i (W/m^2)

Parameters

None

Strings

None

5.2 Block BASE

The BASE block calculates the dynamic conductance value for unsteady heat transport via basement around an building. Calculation is based on EN ISO 13370.

Name	BASE
Function	bs0015
Inputs	0
Outputs	4
Parameters	17
Strings	0
Group	S

Inputs

None

Outputs

1 Dynamic conductance via basement floor [W/K]
2 Dynamic conductance via basement wall [W/K]
3 Static conductance [W/K]
4 Phase shift because of basement [mth.]

Parameters

1 Switch base-mode (0: no cellar without ins. band 1: no cellar with ins. band 2: heated cellar 3: unheated cellar)
2 Switch ins. band modus (only for BP(1)=1) (0: horizontal 1: vertical)
3 Net volume cellar [m^3]
4 Area of basement floor [m^2]
5 External perimeter of basement floor [m]
6 Height of cellar wall [m]
7 Thickness of cellar wall [m]
8 Height cellar wall above ground [m]

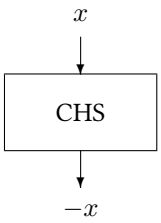
9 Width of insulation band [m]
10 Thickness of ins. band [m]
11 Thermal resistance of wall under ground [(m^2*K)/W]
12 Thermal resistance of ground plate [(m^2*K)/W]
13 U-value cellar ceiling [W/(m^2*K)]
14 Thermal resistance of insulation band (Dämmstreifen) [(m^2*K)/W]
15 U-value wall above ground [W/(m^2*K)
16 Ventilation rate in cellar [1/h]
17 Thermal conductivity of soil [W/(mK)]

Strings

None

5.3 Block CALEND

The CALEND block is a calender handing over a value for profile-use. Every day one profile possible.



Name	CALEND
Function	bs0013
Inputs	2
Outputs	1
Parameters	365 . . . [366]
Strings	0
Group	S

Inputs

1 Month
2 Day of the month

Outputs

1 Integer value

Parameters

1-365 Annula profile

Strings

None

5.4 Block CONZON

The CONZON block determines the thermal resistance between two zones.

Name	CONZON
Function	bs0009
Inputs	0
Outputs	1
Parameters	13
Strings	0
Group	C

Inputs

None

Outputs

1 R_{12}

Parameters

1 DU
2 U1
3 Area of the first kind of wall [m^2]
4 U2
5 Area of the second kind of wall [m^2]
6 U3
7 Area of the third kind of wall [m^2]
8 U4
9 Area of the fourth kind of wall [m^2]
10 U5
11 Area of the fifth kind of wall [m^2]
12 Air change rate of smaler zone [1/h]
13 Net volume of smaler zone [m^3]

Strings

None

5.5 Block CTRLROOM

The CTRLROOM block controls the room air temperature through convective heat fluxes (heating or cooling) onto the air node.

Name	CTRLROOM
Function	bs0028
Inputs	3
Outputs	2
Parameters	9
Strings	0
Group	L

Inputs

1 Room air temperature / °C
2 Setpoint min. room temperature / °C
3 Setpoint max. room temperature / °C

Outputs

1 Convective heating (positive) or cooling (negative) power / W
2 Radiative heating (positive) or cooling (negative) power / W

Parameters

1 Nominal heating power / W
2 Convective/radiative heating power ratio (0: purely radiative, 1: purely convective)
3 Heater type
  0 On/Off
  1 Continuous
  2 TBD
4 Nominal cooling power / W
5 Convective/radiative cooling power ratio (0: purely radiative, 1: purely convective)
6 Cooler type

0 On/Off
1 Continuous
2 TBD

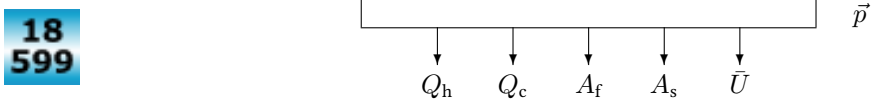
7 Power tolerance / W
8 Maximum number of iterations
9 Failure power / W

Strings

None

5.6 Block D18599

The D18599 block calculates the energy balance of a building for heating and cooling based on DIN 18599.



Name	D18599
Function	bs0014
Inputs	4 . . . [1000]
Outputs	5
Parameters	29 . . . [1000]
Strings	0
Group	S

Inputs

1	Month $\in [1, 12]$
2	Ambient temperature / °C
3	Sky temperature / °C
4	Global irradiation surface 1 / W m^{-2}
..	
3+s	Global irradiation surface s (number of surfaces) / W m^{-2}

Outputs

1	Monthly heating energy demand q_h / kW h
2	Monthly cooling energy demand q_c / kW h
3	Total floor area A_f / m^2
4	Total surface area A_s / m^2
5	Mean U-value $\bar{U}$ / W m^{-2} K^{-1}

Parameters

1	Heated volume
2	Average storey height / m
3	Number of storeys above ground

4 Attic heating type (0=no room, 1=unheated, 2=heated)
5 Cellar heating type (0=no room, 1=unheated, 2=heated)
6 Indirectly heated area ratio
7 Effective heat capacity / $\mathrm{W\,h\,m^{-2}\,K^{-1}}$
8 Additional U-value for heat bridge / $\mathrm{W\,m^{-2}\,K^{-1}}$
9 Usage type (0=standard, 1=IWU)
10 Number z of zones
11 Area A_1 of zone 1 / $\mathrm{m^2}$
12 Internal gains $\dot{q}_\mathrm{i}$ / $\mathrm{W\,m^{-2}}$
13 Heating setpoint temperature / °C
14 Heating setback temperature / °C
15 Cooling setpoint temperature / °C
16 Usage hours per day / h
17 Usage days per year / d
18 Minimum air change rate zone 1 / $\mathrm{h^{-1}}$
19 Area A_2 of zone 2 (optional)
..
26 Minimum air change rate zone 2 (optional)
..
11+8z Number s of surfaces
12+8z Surface 1 type (1=wall, 2=ground, 3= roof, 4=flat roof)
13+8z $A_{\mathrm{a}1}$ Area above ground / m
14+8z $A_{\mathrm{b}1}$ Area below ground / m
16+8z U_1 / $\mathrm{W\,m^{-2}\,K^{-1}}$
16+8z $A_{\mathrm{w}1}$ Window area / $\mathrm{m^2}$
17+8z $F_{\mathrm{r}1}$ Window frame fraction
18+8z $U_{\mathrm{w}1}$ Window U-value / $\mathrm{W\,m^{-2}\,K^{-1}}$
19+8z g_1 Window g-value / $\mathrm{W\,m^{-2}\,K^{-1}}$
20+8z ρ_1 short-wave reflectance of surface 1
21+8z Surface 2 type
..
29+8z ρ_2 short-wave reflectance of surface 2
..

Strings

None

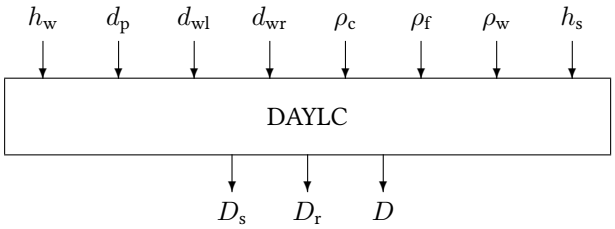
ggfls. aktualisieren !!!

```
s 0 const
p 0 0
s 1 do
p 1 1 12 1 % Months
s 2 polyg 1
p 2 12 % Days per month
%  JAN   FEB   MAR   APR   MAY   JUN   JUL   AUG   SEP   OCT    NOV    DEC
   1 31  2 28  3 31  4 30  5 31  6 30  7 31  8 31  9 30  10 31  11 30  12 31
s 3 gain 2
p 3 24 % Hours per day
s 4 d18599 3 20 21 0 103 104 101 102 100 % Ta Tsky Gr. S_1 S_2 S_3 S_4 Roof
p 4
775       % Heated Volume (vBrutto)
2.5       % Average storey height / m
1         % Number of storeys above ground
0         % Attic heating type (0=no room, 1=unheated, 2=heated)
0         % Cellar heating type (0=no room, 1=unheated, 2=heated)
0.25      % Indirectly heated area ratio
90.0      % Effective heat capacity
0.10      % Additional U-value for heat bridge
1         % Usage type (0=standard, 1=IWU, 99=invald)
1         % Number $z$ of zones
225       % #1 Area of zone 1 (sqm)
13        % #2 Internal gains of zone 1
21        % #3 Heating setpoint temperature zone 1 (tSetHeat)
17        % #4 Heating setback temperature zone 1 (tSetbackHeat)
24        % #5 Cooling setpoint temperature zone 1 (tSetCool)
14        % #6 Usage hours per day zone 1
300       % #7 Usage days per year zone 1
8.3       % #8 Minimum air change rate zone 1 (h^-1)
6         % Number of surfaces = BP(11+8z)
% #1 Surface type (1=wall, 2=ground 3=roof, 4=flat roof), #2 Areas above ground
% #3 Areas below ground, #4 U-value, #5 Window area, #6 Window frame fraction
% #7 Window U-value, #8 Window g-value, #9 Short-wave reflectance
% #1   #2  #3    #4   #5  #6   #7    #8    #9
  2    0  225  0.62   0   0    0     0     0   % groundsurface (beta=0, gamma=n.a.)
  1   45    0  0.83   9   0.3  2.57  0.76  0.3 % wallsurface   (90, 0) N
  1   45    0  0.83   9   0.3  2.57  0.76  0.3 % wallsurface   (90, 90) E
  1   45    0  0.83   9   0.3  2.57  0.76  0.3 % wallsurface   (90, 180) S
  1   45    0  0.83   9   0.3  2.57  0.76  0.3 % wallsurface   (90, 270) W
  4  225    0  0.43   0   0    0     0     0.2 % flat roof     (0, n.a.)
s 20 polyg 1
p 20 12 % Monthly ambient temperature
% JAN   FEB   MAR   APR   MAY    JUN    JUL    AUG    SEP    OCT    NOV    DEC
  1 0.3 2 1.4 3 5.4 4 9.6 5 13.6 6 16.9 7 18.8 8 18.4 9 15.3 10 9.9 11 5.2 12 1.2
s 21 polyg 1
p 21 12 % Monthly sky temperature
% JAN     FEB     MAR     APR    MAY   JUN   JUL   AUG   SEP   OCT     NOV      DEC
  1 -17.3 2 -16.9 3 -10.6 4 -5.7 5 0.2 6 5.3 7 5.7 8 8.3 9 1.5 10 -4.8 11 -10.0 12 -12.4
s 100 polyg 1
p 100 12 % Horizontal Stuttgart
% JAN  FEB  MAR   APR   MAY   JUN    JUL   AUG   SEP   OCT   NOV   DEC
```

```
  1 40 2 69 3 111 4 169 5 207 6 225  7 227 8 187 9 152 10 93 11 46 12 32
s 101 polyg 1
p 101 12 % South
% JAN  FEB  MAR   APR   MAY   JUN   JUL   AUG   SEP   OCT    NOV   DEC
  1 59 2 85 3 103 4 118 5 117 6 117 7 122 8 120 9 131 10 109 11 66 12 51
s 102 polyg 1
p 102 12 % West
% JAN  FEB  MAR  APR   MAY   JUN   JUL   AUG   SEP   OCT   NOV   DEC
  1 28 2 47 3 75 4 105 5 130 6 130 7 136 8 114 9 103 10 61 11 31 12 24
s 103 polyg 1
p 103 12 % North
% JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT   NOV   DEC
  1 19 2 30 3 49 4 71 5 88 6 97 7 97 8 76 9 56 10 37 11 21 12 15
s 104 polyg 1
p 104 12 % East
% JAN  FEB  MAR  APR   MAY   JUN   JUL   AUG   SEP   OCT   NOV   DEC
  1 27 2 44 3 78 4 109 5 129 6 135 7 145 8 121 9 101 10 61 11 32 12 22

s 300 mul 4.1 4.3
s 301 mul 4.2 4.3
s 302 screen 4.1 4.2 300 301
```

5.7 Block DAYLC

The DAYLC block returns the daylight coefficient.



Name	DAYLC
Function	bs0006
Inputs	8
Outputs	3
Parameters	3 . . . [4]
Strings	0
Group	S

Inputs

1 Height h_w of window opening [m]
2 Distance d_p of observer's point from window [m]
3 Distance d_{wl} left to wall [m]
4 Distance d_{wr} right to wall [m]
5 Reflexion coefficient ceiling ρ_c
6 Reflexion coefficient floor ρ_f
7 Reflexion coefficient wall ρ_w
8 Height of shadow from floor in meter h_s

Outputs

1 Skylight part of daylight coefficient D_s
2 Interior reflexion part of daylight coefficient D_r
3 Daylight coefficient D including losses

Parameters

1 Room width W
2 Room depth L
3 Room height H
4 Number of integration steps N

Strings

None

5.8 Block DYBASE

The DYBASE block calculates the dynamic conductance value for unsteady heat transport via basement around an building, the temperature in this basement zone and heating as well as cooling demand. Calculation is based on EN ISO 13370.

Name	DYBASE
Function	bs0011
Inputs	18
Outputs	3
Parameters	36
Strings	0
Group	S

Inputs

1 Second of the year / s
2 Ambient temperature / °C
3 Global radiation on north surface / $W\,m^{-2}$
4 Global radiation on east surface / $W\,m^{-2}$
5 Global radiation on south surface / $W\,m^{-2}$
6 Global radiation on west surface / $W\,m^{-2}$
7 Global radiation on horizontal surface / $W\,m^{-2}$
8 Target temperature / °C
9 Internal gains / W
10 Effective air ventilation rate / h^{-1}
11 Temperature of first adjacent zone / °C
12 Total thermal resistance to first adjacent zone / $K\,W^{-1}$
13 Temperature of second adjacent zone / °C
14 Total thermal resistance to second adjacent zone / $K\,W^{-1}$
15 Temperature of third adjacent zone / °C
16 Total thermal resistance to third adjacent zone / $K\,W^{-1}$

17 Temperature of fourth adjacent zone / °C
18 Total thermal resistance to fourth adjacent zone / $K\,W^{-1}$

Outputs

1 Operative internal temperature / °C
2 Heating energy demand per timestep / kWh
3 Cooling energy demand per timestep / kWh

Parameters

1 Net volume cellar / m^3
2 Area of basement floor / m^2
3 External perimeter of basement floor / m
4 Height of cellar wall / m
5 Thickness of cellar wall / m
6 Thermal resistance of wall under ground / $m^2\,K\,W^{-1}$
7 Thermal resistance of ground plate / $m^2\,K\,W^{-1}$
8 Thermal conductivity of soil / $W\,m^{-1}\,K^{-1}$
9 Additional U-value for heatbridges / $W\,m^{-2}\,K^{-1}$
10 U-value external wall / $W\,m^{-2}\,K^{-1}$
11 Area external wall / m^2
12 Specific thermal capacity / $Wh\,m^{-2}\,K^{-1}$
13 Jahresmitteltemperatur / °C
14 Temperature of initialisation / °C
15 U-value of window north / $W\,m^{-2}\,K^{-1}$
16 Area window north / m^2
17 Total energy transmission of window north
18 Frame fraction of window north / %
19 U-value of window east / $W\,m^{-2}\,K^{-1}$
20 Area window east / m^2
21 Total energy transmission of window east
22 Frame fraction of window east / %
23 U-value of window south / $W\,m^{-2}\,K^{-1}$
24 Area window south / m^2
25 Total energy transmission of window south
26 Frame fraction of window south / %
27 U-value of window west / $W\,m^{-2}\,K^{-1}$
28 Area window west / m^2
29 Total energy transmission of window west
30 Frame fraction of window west / %
31 U-value of window horizontal / $W\,m^{-2}\,K^{-1}$
32 Area window horizontal / m^2
33 Total energy transmission of window horizontal
34 Frame fraction of window horizontal / %
35 Heating on (0) or off (1)

36 Cooling on (0) or off (1)

Strings

None

5.9 Block DYNDGL

The DYNDGL calculates the heating consumption of a building via the numerical solved Differential Equation of a resistant-capacity element. Heat flux through basement is solved with a balance methode.

Name	DYNDGL
Function	bs0018
Inputs	30
Outputs	8
Parameters	5
Strings	0
Group	S

Inputs

1 Second of the year [sec]
2 External ambient temperature [°C]
3 Total irradiation on vertical surface north direction [W/m^2]
4 Total irradiation on vertical surface east direction [W/m^2]
5 Total irradiation on vertical surface south direction [W/m^2]
6 Total irradiation on vertical surface west direction [W/m^2]
7 Total irradiation on horizontal surface [W/m^2]
8 Total thermal resistance outside - zone 1 [K/W]
9 Useable heat capacity zone 1 [h]
10 Factor FF*g*A(windows) north zone 1
11 Factor FF*g*A(windows) east zone 1
12 Factor FF*g*A(windows) south zone 1
13 Factor FF*g*A(windows) west zone 1
14 Factor FF*g*A(windows) horizontal zone 1
15 Target temperature zone 1 [°C]
16 Internal gains zone 1 [W]

17 Total thermal resistance outside - zone 2 [K/W]
18 Useable heat capacity zone 2 [h]
19 Factor FF*g*A(windows) north zone 2
20 Factor FF*g*A(windows) east zone 2
21 Factor FF*g*A(windows) south zone 2
22 Factor FF*g*A(windows) west zone 2
23 Factor FF*g*A(windows) horizontal zone 2
24 Target temperature zone 2 [°C]
25 Internal gains zone 2 [W]
26 Total thermal resistance zone 1 - zone 2 [K/W]
27 Dynamic conductance via basement floor [W/K]
28 Dynamic conductance via basement wall [W/K]
29 Static conductance [W/K]
30 Phase shift because of basement [mth.]

Outputs

1 Heating energy demand per timestep zone 1 [kWh]
2 Cooling energy demand per timestep zone 1 [kWh]
3 Temperature in zone 1 [degC]
4 Heating energy demand per timestep zone 2 [kWh]
5 Cooling energy demand per timestep zone 2 [kWh]
6 Temperature in zone 2 [degC]
7 Heating energy demand per timestep cellar [kWh]
8 ???

Parameters

1 Floor area [m^2]
2 Total ceiling wall area [m^2]
3 Cellar winter setpoint temperature [°C]
4 Cellar summer setpoint temperature [°C]
5 Annual average temperature [°C]

Strings

None

5.10 Block EINZON

The EINZON block determines the thermal parameters of a single zone model.

Name	EINZON
Function	bs0017
Inputs	0
Outputs	8
Parameters	55
Strings	0
Group	S

Inputs

None

Outputs

1 RGES
2 CUSE
3 FN
4 FE
5 FS
6 FW
7 FH
8 ASUR

Parameters

1 DU
2 U1
3 Area of the first kind of wall [m^2]
4 U2
5 Area of the second kind of wall [m^2]
6 U3

7 Area of the third kind of wall [m^2]
8 U4
9 Area of the fourth kind of wall [m^2]
10 U5
11 Area of the fifth kind of wall [m^2]
12 UWN1
13 Area of the first North window [m^2]
14 Total energy transmission of the first North window
15 Frame fraction of the first North window
16 UWN2
17 Area of the second North window [m^2]
18 Total energy transmission of the second North window
19 Frame fraction of the second North window
20 UWE1
21 Area of the first East window [m^2]
22 Total energy transmission of the first East window
23 Frame fraction of the first East window
24 UWE2
25 Area of the second East window [m^2]
26 Total energy transmission of the second East window
27 Frame fraction of the second East window
28 UWS1
29 Area of the first South window [m^2]
30 Total energy transmission of the first South window
31 Frame fraction of the first South window
32 UWS2
33 Area of the second South window [m^2]
34 Total energy transmission of the second South window
35 Frame fraction of the second South window
36 UWW1
37 Area of the first West window [m^2]
38 Total energy transmission of the first West window
39 Frame fraction of the first West window
40 UWW2
41 Area of the second West window [m^2]
42 Total energy transmission of the second West window
43 Frame fraction of the second West window
44 UWH1
45 Area of the first Horizontal window [m^2]
46 Total energy transmission of the first Horizontal window
47 Frame fraction of the first Horizontal window
48 UWH2
49 Area of the second Horizontal window [m^2]

50 Total energy transmission of the second Horizontal window
51 Frame fraction of the second Horizontal window
52 Air change rate [1/h]
53 Net volume of zone [m^3]
54 Specific thermal capacitance
55 Net area [m^2]

Strings

None

5.11 Block ENERBA

The ENERBA (Energy balance) block calculates the energy balance of a building for heating and cooling. Calculation methods are based on DIN V 18599.

Name	ENERBA
Function	bs0100
Inputs	11
Outputs	2
Parameters	70
Strings	0
Group	S

Inputs

1 Day/month of the year [d/month]
2 External ambient temperature [°C]
3 Total irradiation on vertical surface north direction [W/m^2]
4 Total irradiation on vertical surface east direction [W/m^2]
5 Total irradiation on vertical surface south direction [W/m^2]
6 Total irradiation on vertical surface west direction [W/m^2]
7 Total irradiation on horizontal surface [W/m^2]
8 Dynamic conductance via basement floor [W/K]
9 Dynamic conductance via basement wall [W/K]
10 Static conductance [W/K]
11 Phase shift because of basement [mth.]

Outputs

1 Heating energy demand month/day [kWh]
2 Cooling energy demand month/day [kWh]

Parameters

1 Gross volume of the building [m^3]

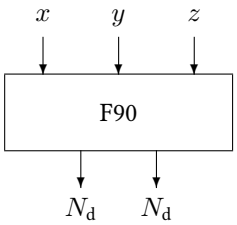
2 Total net area [m^2]
3 Target temperature in winter (for heating) [°C]
4 Target temperature in summer (for cooling) [°C]
5 Calculation modus (0:month 1:day)
6..15 Area of opaque wall 1 to wall 10[m^2]
16 Area of wall under ground level [m^2]
17 Area of bottom plate [m^2]
18 Window area north direction 1 [m^2]
19 Window area north direction 2 [m^2]
20 Window area east direction 1 [m^2]
21 Window area east direction 2 [m^2]
22 Window area south direction 1 [m^2]
23 Window area south direction 2 [m^2]
24 Window area west direction 1 [m^2]
25 Window area west direction 2 [m^2]
26 Window area horizontal 1 [m^2]
27 Window area horizontal 2 [m^2]
28..37 U-value of opaque wall 1 to wall 10 [W/(m^2*K)]
38..47 U-value of window north 1 to 10 [W/(m^2*K)]
48..57 g-value of window north 1 [-]
58 Frame fraction for window group 1 [%]
59 Frame fraction for window group 2 [%]
60 Effective heat capacity [Wh/(m^2*K)]
61 Additional U-value for heat bridge [W/(m^2*K)]
62 ventilation ratio ”blower door” n50 [1/h]
63 ventilation ratio window ventilation [1/h]
64 Modus for night-time heating reduction (0: no nthr 1: nthr 2: shutdown mode)
65 Lenght of night-time heating reduction [h]
66 Maximum temperature reduction during night [K]
67 Time of use (for example 10) [h]
68 Internal gains per area during useing [W/m^2]
69 Internal gains per area off useing [W/m^2]
70 Average temperature during year [°C]

Strings

None

5.12 Block F90

The F90 block calculates the radiation shape factor of two perpendicular rectangles with a common edge.



Name	F90
Function	bs0007
Inputs	3
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1 x-coordinate of A_1 and A_2 (common edge)
2 y-coordinate of A_1 (sender)
3 z-coordinate of A_2 (receiver)

Outputs

1 Radiation shape factor $F_{1\to2}$
2 Radiation shape factor $F_{2\to1}$

Parameters

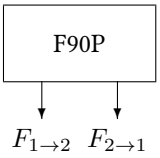
None

Strings

None

5.13 Block F90P

The F90P block calculates the radiation shape factor of two perpendicular rectangles without a common edge.



Name	F90P
Function	bs0019
Inputs	0
Outputs	2
Parameters	9
Strings	0
Group	C

Inputs

None

Outputs

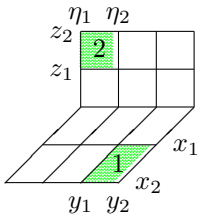
1 Radiation shape factor $F_{1\to2}$
2 Radiation shape factor $F_{2\to1}$

Parameters

1 x-coordinate displacement x_1 of area 1 (sender)
2 x_2 coordinate of area 1 (width $+x_1$)
3 y-coordinate displacement y_1 of area 1 (sender)
4 y_2 coordinate of area 1 (depth $+y_1$)
5 x-coordinate displacement x_1 of area 2 (receiver)
6 x_2 coordinate of area 2 (width $+x_1$)
7 z-coordinate displacement z_1 of area 2 (receiver)
8 z_2 coordinate of area 2 (height $+z_1$)

Strings

None



According to Howell the governing equation is

$$F_{1\to 2} = \frac{1}{2\pi(x_2 - x_1)(y_2 - y_1)} \sum_{l=1}^{2}\sum_{k=1}^{2}\sum_{j=1}^{2}\sum_{i=1}^{2}(-1)^{i+j+k+l} G(x_i, y_j, \eta_k, z_l)$$

where

$$\begin{aligned} G =& (y-\eta)\sqrt{x^2+z^2}\arctan\left(\frac{y-\eta}{\sqrt{x^2+z^2}}\right) \\ & - \frac{1}{4}\left(x^2+z^2-(y-\eta)^2\right)\ln(x^2+z^2+(y-\eta)^2) \end{aligned}$$

5.14 Block FACADE

The FACADE block calculates the outlet temperature of...

Name	FACADE
Function	bs0020
Inputs	7
Outputs	1
Parameters	1
Strings	0
Group	S

Inputs

1	Tin
2	T1
3	T2
4	hc1
5	hc2
6	v
7	x

Outputs

1	T(x)

Parameters

1	Gap width s

Strings

None

5.15 Block FADYBS

The FADYBS block calculates the heating consumption of a zone via the numerical solved differential equation of a resistant-capacity element. FADYBS is short for Fast Dynamic Building Simulation. The FADYBS block is deprecated. Its sucessor is block FDBS.

Name	FADYBS
Function	bs0010
Inputs	18
Outputs	3
Parameters	55 . . . [57]
Strings	0
Group	S

Inputs

1 Second of the year / s
2 External ambient temperature / °C
3 Global radiation on vertical surface north direction / $W\,m^{-2}$
4 Global radiation on vertical surface east direction / $W\,m^{-2}$
5 Global radiation on vertical surface south direction / $W\,m^{-2}$
6 Global radiation on vertical surface west direction / $W\,m^{-2}$
7 Global radiation on horizontal surface / $W\,m^{-2}$
8 Target temperature / °C
9 Internal gains / W
10 Effective air ventilation rate / h^{-1}
11 Temperature of first adjacent zone / °C
12 Total thermal resistance to first adjacent zone / $K\,W^{-1}$
13 Temperature of second adjacent zone / °C
14 Total thermal resistance to second adjacent zone / $K\,W^{-1}$
15 Temperature of third adjacent zone / °C

16 Total thermal resistance to third adjacent zone / $K\,W^{-1}$
17 Temperature of fourth adjacent zone / °C
18 Total thermal resistance to fourth adjacent zone / $K\,W^{-1}$

Outputs

1 Operative internal temperature / °C
2 Heating energy demand per timestep / kWh
3 Cooling energy demand per timestep / kWh

Parameters

1 Additional U-value for heatbridges / $W\,m^{-2}\,K^{-1}$
2 U-value external wall 1 / $W\,m^{-2}\,K^{-1}$
3 Area external wall 1 / m^2
4 U-value external wall 2 / $W\,m^{-2}\,K^{-1}$
5 Area external wall 2 / m^2
6 U-value external wall 3 / $W\,m^{-2}\,K^{-1}$
7 Area external wall 3 / m^2
8 U-value external wall 4 / $W\,m^{-2}\,K^{-1}$
9 Area external wall 4 / m^2
10 U-value external wall 5 / $W\,m^{-2}\,K^{-1}$
11 Area external wall 5 / m^2
12 U-value of window north 1 / $W\,m^{-2}\,K^{-1}$
13 Area window north 1 / m^2
14 Total energy transmission of window north 1
15 Frame fraction of window north 1 / %
16 U-value of window north 2 / $W\,m^{-2}\,K^{-1}$
17 Area window north 2 / m^2
18 Total energy transmission of window north 2
19 Frame fraction of window north 2 / %
20 U-value of window east 1 / $W\,m^{-2}\,K^{-1}$
21 Area window east 1 / m^2
22 Total energy transmission of window east 1
23 Frame fraction of window east 1 / %
24 U-value of window east 2 / $W\,m^{-2}\,K^{-1}$
25 Area window east 2 / m^2
26 Total energy transmission of window east 2
27 Frame fraction of window east 2 / %
28 U-value of window south 1 / $W\,m^{-2}\,K^{-1}$
29 Area window south 1 / m^2
30 Total energy transmission of window south 1
31 Frame fraction of window south 1 / %
32 U-value of window south 2 / $W\,m^{-2}\,K^{-1}$
33 Area window south 2 / m^2
34 Total energy transmission of window south 2

35 Frame fraction of window south 2 / %
36 U-value of window west 1 / $W\,m^{-2}\,K^{-1}$
37 Area window west 1 / m^2
38 Total energy transmission of window west 1
39 Frame fraction of window west 1 / %
40 U-value of window west 2 / $W\,m^{-2}\,K^{-1}$
41 Area window west 2 / m^2
42 Total energy transmission of window west 2
43 Frame fraction of window west 2 / %
44 U-value of window horizontal 1 / $W\,m^{-2}\,K^{-1}$
45 Area window horizontal 1 / m^2
46 Total energy transmission of window horizontal 1
47 Frame fraction of window horizontal 1 / %
48 U-value of window horizontal 2 / $W\,m^{-2}\,K^{-1}$
49 Area window horizontal 2 / m^2
50 Total energy transmission of window horizontal 2
51 Frame fraction of window horizontal 2 / %
52 Net volume of zone (Air-volume) / m^3
53 Net area / useful area / m^2
54 Specific thermal capacity / $W\,m^{-2}\,K^{-1}$
55 Temperature of initialisation / °C
56 Heating on (0) or off (1)
57 Cooling on (0) or off (1)

None

5.16 Block FDBS

The FDBS block calculates the heating consumption of a zone via the numerical solved differential equation of a resistant-capacity element. FDBS is short for Fast Dynamic Building Simulation.

Name	FDBS
Function	bs0024
Inputs	18
Outputs	3
Parameters	55 . . . [57]
Strings	0
Group	S

Inputs

1 Second of the year / s
2 External ambient temperature / °C
3 Global radiation on vertical surface north direction / $W\,m^{-2}$
4 Global radiation on vertical surface east direction / $W\,m^{-2}$
5 Global radiation on vertical surface south direction / $W\,m^{-2}$
6 Global radiation on vertical surface west direction / $W\,m^{-2}$
7 Global radiation on horizontal surface / $W\,m^{-2}$
8 Target temperature / °C
9 Internal gains / W
10 Effective air ventilation rate / h^{-1}
11 Temperature of first adjacent zone / °C
12 Total thermal resistance to first adjacent zone / $K\,W^{-1}$
13 Temperature of second adjacent zone / °C
14 Total thermal resistance to second adjacent zone / $K\,W^{-1}$
15 Temperature of third adjacent zone / °C
16 Total thermal resistance to third adjacent zone / $K\,W^{-1}$

17 Temperature of fourth adjacent zone / °C
18 Total thermal resistance to fourth adjacent zone / $K\,W^{-1}$

Outputs

1 Operative internal temperature / °C
2 Heating energy demand per timestep / kWh
3 Cooling energy demand per timestep / kWh

Parameters

52 Net zone volume (air-volume) / m^3
53 Floor area / useful area / m^2
3 Area external wall 1 / m^2
2 U-value external wall 1 / $W\,m^{-2}\,K^{-1}$
13 Area window north 1 / m^2
12 U-value of window north 1 / $W\,m^{-2}\,K^{-1}$
5 Area external wall 2 / m^2
4 U-value external wall 2 / $W\,m^{-2}\,K^{-1}$
17 Area window north 2 / m^2
16 U-value of window north 2 / $W\,m^{-2}\,K^{-1}$
7 Area external wall 3 / m^2
6 U-value external wall 3 / $W\,m^{-2}\,K^{-1}$
9 Area external wall 4 / m^2
8 U-value external wall 4 / $W\,m^{-2}\,K^{-1}$
11 Area external wall 5 / m^2
10 U-value external wall 5 / $W\,m^{-2}\,K^{-1}$
14 Total energy transmission of window north 1
15 Frame fraction of window north 1 / %
18 Total energy transmission of window north 2
19 Frame fraction of window north 2 / %
20 U-value of window east 1 / $W\,m^{-2}\,K^{-1}$
22 Total energy transmission of window east 1
24 U-value of window east 2 / $W\,m^{-2}\,K^{-1}$
25 Area window east 2 / m^2
26 Total energy transmission of window east 2
27 Frame fraction of window east 2 / %
28 U-value of window south 1 / $W\,m^{-2}\,K^{-1}$
29 Area window south 1 / m^2
30 Total energy transmission of window south 1
31 Frame fraction of window south 1 / %
32 U-value of window south 2 / $W\,m^{-2}\,K^{-1}$
33 Area window south 2 / m^2
34 Total energy transmission of window south 2
35 Frame fraction of window south 2 / %
36 U-value of window west 1 / $W\,m^{-2}\,K^{-1}$

37 Area window west 1 / m^2
38 Total energy transmission of window west 1
39 Frame fraction of window west 1 / %
40 U-value of window west 2 / $W\,m^{-2}\,K^{-1}$
41 Area window west 2 / m^2
42 Total energy transmission of window west 2
43 Frame fraction of window west 2 / %
44 U-value of window horizontal 1 / $W\,m^{-2}\,K^{-1}$
45 Area window horizontal 1 / m^2
46 Total energy transmission of window horizontal 1
47 Frame fraction of window horizontal 1 / %
48 U-value of window horizontal 2 / $W\,m^{-2}\,K^{-1}$
49 Area window horizontal 2 / m^2
50 Total energy transmission of window horizontal 2
51 Frame fraction of window horizontal 2 / %
54 Specific thermal capacity / $W\,m^{-2}\,K^{-1}$
55 Initial temperature / °C
56 Heating on (0) or off (1)
57 Cooling on (0) or off (1)

5.17 Block FIT3PC

The FIT3PC block approximates the input data by a three-parameter model typically used for cooling demand calculations of buildings.

FIT3PC

Name	FIT3PC
Function	bs0031
Inputs	2
Outputs	4
Parameters	0
Strings	0
Group	I

Inputs

1 Temperature T_a / °C
2 Demand / kWh

Outputs

1 Cooling base load Q_{cb} / kWh
2 Cooling change-point temperature T_c / °C
3 Slope α_c / kWh K^{-1}
4 Regression coefficient r^2

Parameters

None

Strings

None

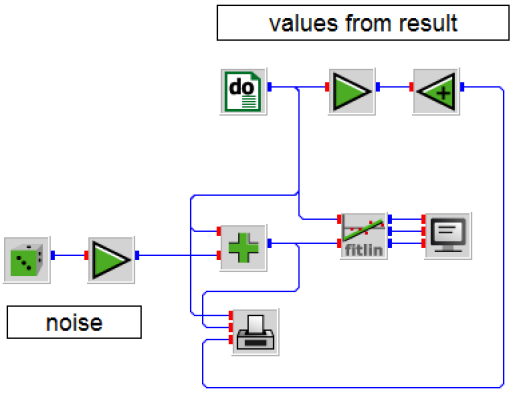
Description The blocks input data are approximated by the linear equation

$$Q_c = Q_{cb} + \alpha_c (T_a - T_c)^+$$

With the notation $(T_1 - T_2)^+$ where temperature differences are only accounted for if positive and zero otherwise The successors of this block are executed only at the end of the simulation run.

BEISPIEL ANPASSEN

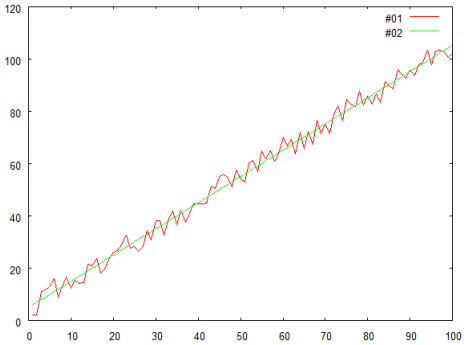
fitlin.vseit



A DO block delivers 100 clicks, i. e., counts from 1 to 100. A RAN1 block generates one uniformly distributed random number for each step multiplied by 10 by a GAIN block. A SUM block adds the DO block's output and the output of the GAIN block giving the variable to be fitted as a function of the DO block's output by the FITLIN block. The outputs a, b, and r^2 are then displayed numerically by a SCREEN block.

```
4.8319  1.0039  0.9903
```

Assuming that the result is already known ($a = 4.8319$ and $b = 1.0039$) the linear equation $y' = a + bx$ is calculated. A PLOT block displays the data and the fit on screen.



Remarks The FITLIN block is a typical example for an I-block: Its successors are executed (once) only when a certain condition is fullfilled—here the end of the main simulation loop. Compare this behavior with a typical example for a T-block like block FDIST, for example.

5.18 Block FIT3PH

The FIT3PH block approximates the input data by a three-parameter model typically used for heating-demand calculations of buildings.

FIT3PH

Name	FIT3PH
Function	bs0031
Inputs	2
Outputs	4
Parameters	0
Strings	0
Group	I

Inputs

1 Temperature T_a / °C
2 Demand / kWh

Outputs

1 Heating base load Q_{hb} / kWh
2 Heating change-point temperature T_h / °C
3 Slope α_h / kWh K^{-1}
4 Regression coefficient r^2

Parameters

None

Strings

None

5.19 Block FIT4PC

The FIT4PC block approximates the input data by a four-parameter model typically used for cooling demand calculations of buildings.

FIT4PC

Name	FIT4PC
Function	bs0031
Inputs	2
Outputs	5
Parameters	0
Strings	0
Group	I

Inputs

1 Temperature T_a / °C
2 Demand / kWh

Outputs

1 Cooling base load Q_{cb} / kWh
2 Cooling change-point temperature T_c / °C
3 Cooling slope α_c / kWh K^{-1} (usually negative)
4 Baseload slope α_{cb} / kWh K^{-1} (usually negative)
5 Regression coefficient r^2

Parameters

None

Strings

None

5.20 Block FIT4PH

The FIT4PH block approximates the input data by a four-parameter model typically used for heating-demand calculations of buildings.

FIT4PH

Name	FIT4PH
Function	bs0031
Inputs	2
Outputs	5
Parameters	0
Strings	0
Group	I

Inputs

1 Temperature T_a / °C
2 Demand / kWh

Outputs

1 Heating base load Q_{hb} / kWh
2 Heating change-point temperature T_h / °C
3 Heating slope α_h / kWh K^{-1} (usually negative)
4 Baseload slope α_{hb} / kWh K^{-1} (usually negative)
5 Regression coefficient r^2

Parameters

None

Strings

None

5.21 Block FIT5PH

The FIT5P block approximates the input data by a five-parameter model typically used for heating- and cooling-demand calculations of buildings.

Name	FIT5PH
Function	bs0032
Inputs	2
Outputs	4
Parameters	0
Strings	0
Group	I

Inputs

1 Temperature T_a / °C
2 Demand / kWh

5.22 Block FPARA

The FPARA block calculates the radiation shape factor of two parallel rectangles.

FPARA

Name	FPARA
Function	bs0007
Inputs	3
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1 x-coordinate of A_1 and A_2
2 y-coordinate of A_1 (sender) and A_2
3 z-coordinate of A_2 (receiver)

Outputs

1 Radiation shape factor $F_{1\to 2}$
2 Radiation shape factor $F_{2\to 1}$

Parameters

None

Strings

None

5.23 Block FPARAP

The F90P block calculates the radiation shape factor of two parallel displaced rectangles without a common edge.

FPARAP

Name	FPARAP
Function	bs0019
Inputs	0
Outputs	2
Parameters	9
Strings	0
Group	C

Inputs

None

Outputs

1 Radiation shape factor $F_{1\to 2}$
2 Radiation shape factor $F_{2\to 1}$

Parameters

1 x-coordinate displacement x_1 of area 1 (sender)
2 x_2 coordinate of area 1 (width $+x_1$)
3 y-coordinate displacement y_1 of area 1 (sender)
4 y_2 coordinate of area 1 (depth $+y_1$)
5 x-coordinate displacement x_1 of area 2 (receiver)
6 x_2 coordinate of area 2 (width $+x_1$)
7 y-coordinate displacement η_1 of area 2 (receiver)
8 η_2 coordinate of area 2 (depth $+\eta_1$)
9 Distance z between the parallel planes

Strings

None

5.24 Block GMIX

The GMIX block distributes heat flux and beam and diffuse radiation over six walls.

Name	GMIX
Function	bs0023
Inputs	3 . . . 5
Outputs	6
Parameters	3 . . . 7
Strings	0
Group	S

Inputs

1 Heat flux (excluding radiation) / W
2 Beam radiation / W
3 Diffuse radiation / W
4 Deviation from normal azimuth / degrees
5 Effective elevation / degrees

Outputs

1 Global heat flux to opposing wall / $\mathrm{W\,m^{-2}}$
2 Global heat flux to right wall / $\mathrm{W\,m^{-2}}$
3 Global heat flux to self / $\mathrm{W\,m^{-2}}$
4 Global heat flux to left wall / $\mathrm{W\,m^{-2}}$
5 Global heat flux to floor / $\mathrm{W\,m^{-2}}$
6 Global heat flux to ceiling / $\mathrm{W\,m^{-2}}$

Parameters

1 Width of wall through which heat flux enters / m
2 Distance to opposing wall / m
3 Room height / m
4 x-coordinate of lower left glas corner

5 y-coordinate of lower left glas corner
6 Glas width / m
7 Glas height / m

Strings

None

5.25 Block HR

The HR block calculates the radiative heat transfer coefficient for two surfaces with a view factor $F_{1\to 2}$.

Name	HR
Function	bs0021
Inputs	3
Outputs	1
Parameters	4
Strings	0
Group	S

Inputs

1	T1
2	T2
3	F12

Outputs

1	hr

Parameters

1	A1
2	A2
3	Emissivity of surface one
4	Emissivity of surface two

Strings

None

5.26 Block IRADIA

The IRADIA block calculates the distribution of solar radiation on the surface of internal walls

Name	IRADIA
Function	bs0025
Inputs	5 + nwalls
Outputs	nwalls + 1
Parameters	nwalls + 3
Strings	0
Group	S

Inputs

1	Total solar irradiation through all windows / W

Outputs

i	..OUT(nwalls) specific solar radiation heat flux to surface node i / W m^{-2}

Parameters

1	Number of walls n
i+1	Surface area of wall i / m^2
n+1+i	Short wave absorption coefficient α_i for windows (alpha-window + Tau-win(60 deg))

Strings

None

5.27 Block Q3PC

The Q3PC block implements a three-parameter model typically used for cooling demand calculations of buildings.

Q3PC

Name	Q3PC
Function	bs0030
Inputs	1
Outputs	1
Parameters	3
Strings	0
Group	S

Inputs

1 Ambient temperature T_a / °C

Outputs

1 Cooling demand Q_c / kWh

Parameters

1 Cooling base load Q_{cb} / kWh
2 Cooling change-point temperature T_c / °C
3 Cooling slope α_c / kWh K^{-1} (usually negative)

Strings

None

5.28 Block Q3PH

The Q3PH block implements by a three-parameter model typically used for heating-demand calculations of buildings.

Q3PH

Name	Q3PH
Function	bs0030
Inputs	1
Outputs	1
Parameters	3
Strings	0
Group	S

Inputs

1 Ambient temperature T_a / °C

Outputs

1 Heating demand Q_h / kWh

Parameters

1 Heating base load Q_{hb} / kWh
2 Heating change-point temperature / °C
3 Heating slope α_h / kWh K^{-1} (usually negative)

Strings

None

5.29 Block Q4PC

The Q4PC block implements a four-parameter model typically used for cooling demand calculations of buildings.

Q4PC

Name	Q4PC
Function	bs0030
Inputs	1
Outputs	1
Parameters	4
Strings	0
Group	S

Inputs

1 Ambient temperature T_a / °C

Outputs

1 Cooling demand Q_c / kWh

Parameters

1 Cooling base load Q_{cb} / kWh
2 Cooling change-point temperature T_c / °C
3 Cooling slope α_c / kWh K^{-1} (usually negative)
4 Baseload slope α_{cb} / kWh K^{-1} (usually negative)

Strings

None

5.30 Block Q4PH

The Q4PH block implements by a four-parameter model typically used for heating-demand calculations of buildings.

Q4PH

Name	Q4PH
Function	bs0030
Inputs	1
Outputs	1
Parameters	4
Strings	0
Group	S

Inputs

1 Ambient temperature T_a / °C

Outputs

1 Heating demand Q_h / kWh

Parameters

1 Heating base load Q_{hb} / kWh
2 Heating change-point temperature T_h / °C
3 Heating slope α_h / kWh K^{-1} (usually negative)
4 Baseload slope α_{hb} / kWh K^{-1} (usually negative)

Strings

None

5.31 Block Q5P

The Q5P block implements by a five-parameter model typically used for heating- and cooling demand calculations of buildings.

Q5P

Name	Q5P
Function	bs0029
Inputs	1
Outputs	1
Parameters	5
Strings	0
Group	S

Inputs

1 Ambient temperature T_a / °C

5.32 Block QDTV

The QDTV block calculates the heat flux to or from an external wall surface due to free and forced convection of the amient air.

Name	QDTV
Function	bs0027
Inputs	4
Outputs	1
Parameters	1 . . . [2]
Strings	0
Group	S

Inputs

1 Surface temperature of the wall T_w / °C
2 Ambient temperature T_a / °C
3 Wind speed v_w / m s^{-1}
4 Relative wind direction $v_{dir,r}$ (0 ° = normal, 90 ° = parallel)

Outputs

1 Convective heat flux from ambient air $\dot{q}_a$

Parameters

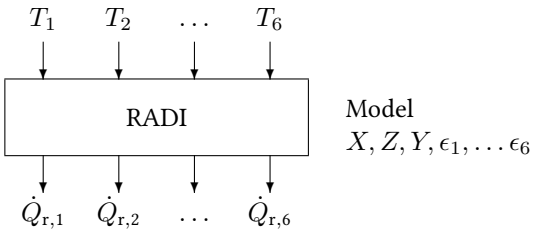
1 Model
  0 Constant value
  1 Third-power average between free and forced convection
2 Heat transfer coefficient value h_c (model 0 only)

Strings

None

5.33 Block RADI

The RADI block calculates the long wave heat exchange between surface areas of different temperatures.



Name	RADI
Function	bs0003
Inputs	6
Outputs	6
Parameters	10
Strings	0
Group	S

Inputs

1 Surface temperature of south wall / °C
2 Surface temperature of west wall / °C
3 Surface temperature of north wall / °C
4 Surface temperature of east wall / °C
5 Surface temperature of ceiling / °C
6 Surface temperature of floor / °C

Outputs

1 Radiative heat flux to south wall / $\mathrm{W\,m^{-2}}$
2 Radiative heat flux to west wall / $\mathrm{W\,m^{-2}}$
3 Radiative heat flux to north wall / $\mathrm{W\,m^{-2}}$
4 Radiative heat flux to east wall / $\mathrm{W\,m^{-2}}$
5 Radiative heat flux to ceiling / $\mathrm{W\,m^{-2}}$
6 Radiative heat flux to floor / $\mathrm{W\,m^{-2}}$

Parameters

1 Model
0 NN

1 Glück
2 VDI 6020

2 Size of the room in x-direction (west-east) / m
3 Size of the room in y-direction (north-south) / m
4 Size of the room in z-direction (height) / m
5..10 Emission coefficients of each wall (same order as inputs and outputs)

Strings

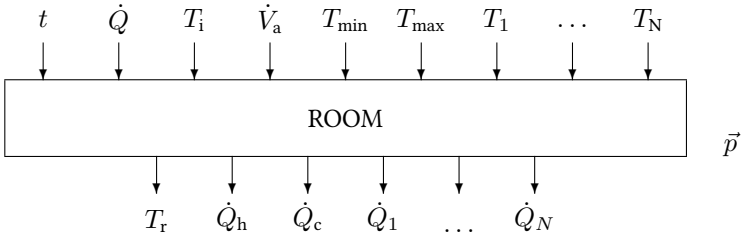
None

Remarks

In general, the radiative heat exchange has to be calculated considering the emission of each surface in direction of another surface, the reflection of that surface back to the original and finally the re-reflexion of the original surface. To simplify the calculations, the net longwave emission of each surface is calculated from the own emission of the surface reduced by the absorbed irradiance at the surface, which comes from all the surrounding surfaces. For the surrounding surface intensities, their own emission is calculated and the reflection of a mean room radiation intensity is added.

5.34 Block ROOM

The ROOM block calculates the room air temperature from the convective heat fluxes (heating or cooling) onto the air node. The output of the room block is the room air temperature at the end of the timestep, the requested heating or cooling demand, and the convective heat flux to each surface temperature node.



Name	ROOM
Function	bs0002
Inputs	$6 + N$
Outputs	$3 + N$
Parameters	$3 + 2N$
Strings	0
Group	S

Inputs

1	Time / s
2	Convective power on air node / W
3	Ventilation air temperature / °C
4	Ventilation air exchange rate / h^{-1}
5	Setpoint min. room temperature / °C
6	Setpoint max. room temperature / °C
6+i	..IN(6+N) Surface temperature of wall i / °C

Outputs

1	Room air temperature / °C
2	Requested heating power / W
3	Requested cooling power / W
3+i	..OUT(3+N) Specific convective heat flux to surface node i / $\mathrm{W\,m}^{-2}$

Parameters

1	Room air volume / m^3

2	Initial value of air temperature / °C
3	Number of walls $N \in [1, 20]$. The order of the wall parameters must be identical to the order of the wall temperature inputs. It is recommended to use the same ordering as with the RADI block i. e., south, west, north etc. or the RADIN block.
2+2i	Surface area of wall i / m^2
3+2i	Heat transfer coefficient of wall i / $W\,m^{-2}\,K^{-1}$

Strings

None

Description

The model of the ROOM block is basically an energy balance between the convective heat input plus the ventilation gains and the induced heat fluxes which lead to an increse of the air temperature and to heat fluxes to the surrounding walls

$$\dot{Q}_c = \frac{\rho V c_p}{\Delta t}(T - T_{old}) + \sum_{i=1}^{N} h_i A_i (T - T_i)$$

When the ventilation rate is greater than one volume per time step the time step is reduced inside the ROOM block. In the first step one volume is exchanged, this results in a step with of

$$\Delta t =$$

room.vseit

5.35 Block SCHEDU

The SCHEDU block determines the current value, e.g. for internal gains or aim-temperature, according to a time-schedule for days during the week and on weekend.

Name	SCHEDU
Function	bs0012
Inputs	2
Outputs	2
Parameters	54
Strings	0
Group	S

Inputs

1 Integer nr. from calendar-block [-]
2 Minute of the day [min]

Outputs

1 Current value [e.g. W, degC]

Parameters

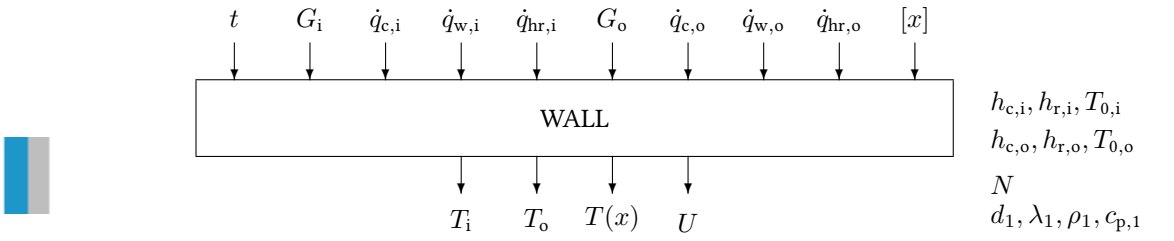
1+3*n From Time [h]
2+3*n Until Time [h]
3+3*n Value

Strings

None

5.36 Block WALL

The WALL block solves the one-dimensional heat transfer equation for an interior wall consisting of several layers with varying thickness and heat conductivity. Shortwave irradiance, convective heat flux from the room air and longwave radiation flux from the other surfaces are given as boundary conditions.



Name	WALL
Function	bs0001
Inputs	9 . . . [10]
Outputs	4
Parameters	11 . . . [47]
Strings	0
Group	3

Inputs

1 Time / s
2 Interior solar radiation gains G_i / W m^{-2}
3 Interior convective heat flux from room air $\dot{q}_\mathrm{c,i}$ / W m^{-2}
4 Longwave heat flux from interior wall surfaces $\dot{q}_\mathrm{w,i}$ / W m^{-2}
5 Radiative part of internal heat sources $\dot{q}_\mathrm{hr,i}$ / W m^{-2}
6 Exterior solar radiation gains G_o / W m^{-2}
7 Exterior convective heat flux from room air $\dot{q}_\mathrm{c,o}$ / W m^{-2}
8 Longwave heat flux from exterior wall surfaces $\dot{q}_\mathrm{w,o}$ / W m^{-2}
9 Radiative part of exterior heat sources $\dot{q}_\mathrm{hr,o}$ / W m^{-2}
10 Distance x from inner surface / m

Outputs

1 Interior surface temperature at $t + \Delta t$
2 Exterior surface temperature at $t + \Delta t$
3 Temperature at distance x from inner surface
4 U-value / W m^{-2} K^{-1}

Parameters

1	Interior convective heat transfer coefficient $h_{c,i}$ / $W\,m^{-2}\,K^{-1}$
2	Interior radiative heat transfer coefficient $h_{r,i}$ / $W\,m^{-2}\,K^{-1}$
3	Initial value interior air temperature $T_{0,i}$ / °C
4	Exterior convective heat transfer coefficient $h_{c,o}$ / $W\,m^{-2}\,K^{-1}$
5	Exterior radiative heat transfer coefficient $h_{r,o}$ / $W\,m^{-2}\,K^{-1}$
6	Initial value exterior air temperature $T_{0,o}$ / °C
7	Number of different material layers of wall $1 \leq N_l \leq 10$
8	Thickness of material layer one / m
9	Heat conductivity of layer one / $W\,m^{-1}\,K^{-1}$
10	Density of material layer one / $kg\,m^{-3}$
11	Heat capacity of material layer one / $J\,kg^{-1}\,K^{-1}$
12	Thickness of further material layer
13	Heat conductivity of further layer
14	Density of further material layer
15	Heat capacity of further material layer
...	etc.

Strings

None

Description

The model of the WALL block uses a finite difference scheme to solve the heat transport and temperature distribution inside the wall.

The heat flux to the two internal wall surfaces is the sum of the solar gains G, the convective heat flux from the room air on each side $\dot{q}_c$, the long-wave radiation from the surrounding wall surfaces $\dot{q}_w$, and the radiative part of internal heating (or cooling) gains $\dot{q}_{hr}$

$$\dot{q} = G + \dot{q}_c + \dot{q}_w + \dot{q}_{hr}$$

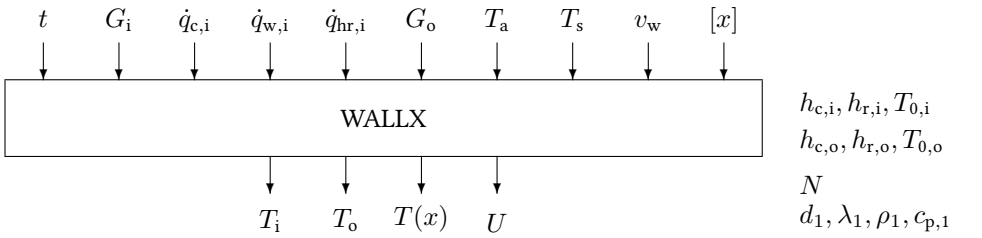
These heat fluxes are absorbed by the boundary layers of the wall and lead to a temperature increase of those layers. Depending on the temperature of the neighboring layers a heat flux is transferred to or from the neighboring layer by conduction. This heat flux depends only on the temperature difference between those layers and is given as

()

wall.vseit

5.37 Block WALLX

The WALLX block solves the one-dimensional heat transfer equation for an external wall consisting of several layers with varying thickness and heat conductivity. As boundary conditions, the exterior temperature and solar irradiance is given for the outside wall surface, on the inside shortwave irradiance, convective heat flux from the room air and longwave radiation flux from the other surfaces are given as boundary conditions.



Name	WALLX
Function	bs0001
Inputs	9 . . . [10]
Outputs	4
Parameters	11 . . . [47]
Strings	0
Group	3

Inputs

1 Time / s
2 Interior solar radiation gains G_i / W m^{-2}
3 Interior convective heat flux from room air $\dot{q}_\mathrm{c,i}$ / W m^{-2}
4 Longwave heat flux from interior wall surfaces $\dot{q}_\mathrm{w,i}$ / W m^{-2}
5 Radiative part of internal heat sources $\dot{q}_\mathrm{hr,i}$ / W m^{-2}
6 Exterior solar radiation gains / W m^{-2} G_o / W m^{-2}
7 Ambient temperature / °C
8 Sky temperature / °C
9 Wind speed / m s^{-1}
6 Exterior solar radiation gains / W m^{-2} G_o / W m^{-2}
7 Exterior convective heat flux from ambient air $\dot{q}_\mathrm{c,o}$ / W m^{-2}
8 Exterior radiative heat flux from sky $\dot{q}_\mathrm{rs,o}$ / W m^{-2}
9 Exterior radiative heat flux from environment $\dot{q}_\mathrm{re,o}$ / W m^{-2}
10 Distance x from inner surface / m

Outputs

1	Interior surface temperature at $t + \Delta t$
2	Exterior surface temperature at $t + \Delta t$
3	Temperature at distance x from inner surface
4	U-value / $\mathrm{W\,m^{-2}\,K^{-1}}$

Parameters

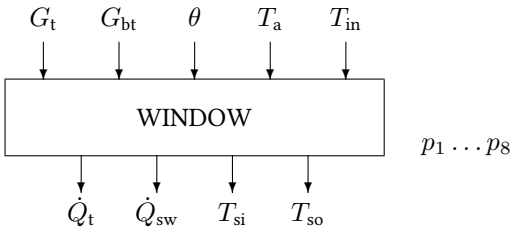
1	Interior convective heat transfer coefficient $h_{\mathrm{c,i}}$ / $\mathrm{W\,m^{-2}\,K^{-1}}$
2	Interior radiative heat transfer coefficient $h_{\mathrm{r,i}}$ / $\mathrm{W\,m^{-2}\,K^{-1}}$
3	Initial value interior air temperature $T_{0,\mathrm{i}}$ / °C
4	Exterior convective heat transfer coefficient $h_{\mathrm{c,o}}$ / $\mathrm{W\,m^{-2}\,K^{-1}}$
5	Exterior radiative heat transfer coefficient $h_{\mathrm{r,o}}$ / $\mathrm{W\,m^{-2}\,K^{-1}}$
6	Initial value exterior air temperature $T_{0,\mathrm{o}}$ / °C
7	Number of different material layers of wall $1 \leq N_{\mathrm{l}} \leq 10$
8	Thickness of material layer one / m
9	Heat conductivity of layer one / $\mathrm{W\,m^{-1}\,K^{-1}}$
10	Density of material layer one / $\mathrm{kg\,m^{-3}}$
11	Heat capacity of material layer one / $\mathrm{J\,kg^{-1}\,K^{-1}}$
12	Thickness of further material layer
13	Heat conductivity of further layer
14	Density of further material layer
15	Heat capacity of further material layer
...	etc.

Strings

None

5.38 Block WINDOW

The WINDOW block calculates the total energy flux through a window, with a detailed model of the solar heat gain coefficient.



Name	WINDOW
Function	bs0008
Inputs	5
Outputs	5
Parameters	8
Strings	0
Group	S

Inputs

1 Global radiation in the window plane G_t / W m^{-2}
2 Beam radiation in the window plane G_{bt} / W m^{-2}
3 Incidence angle θ of the beam radiation over the window in degrees
4 Ambient air temperature T_a / °C
5 Inside air temperature T_{in} / °C
6 Radiative heat flux inside

Outputs

1 Total heat flux through the window, from outside to inside $\dot{Q}_t$ / W
2 Beam radiation transmitted through the window $\dot{Q}_b$ / W
3 Diffuse radiation transmitted through the window $\dot{Q}_d$ / W
4 Inside surface temperature T_{si} / °C
5 Outside surface temperature T_{so} / °C

Parameters

1 Global U-value including window frame / W m^{-2} K^{-1}
2 Number of window panes N_p
3 Total window Area A / m^2

4	Frame fraction of the total window area
5	Optical absorption coefficient α of inner glass pane
6	Normal solar heat gain coefficient, without shading $SHGC_n$
7	Normal solar heat gain coefficient, with shading $SHGC_{sh}$
8	Global radiation threshold above which one the shading devices are set on G_{th} / $W\,m^{-2}$

Strings

None

Remarks

The WINDOW block calculates the total heat flux through the window (solar gains and transmission losses), the short wave radiation transmitted through the window, and both surface temperatures.

6 :: Solar Power

6.1 Block CNTRL

The CNTRL block...

Name	CNTRL
Function	pp0010
Inputs	3
Outputs	1
Parameters	6
Strings	0
Group	S

Inputs

1 Collector mass flow
2 Collector outlet temperature
3 Collector inlet temperature

Outputs

1 New mass flow

Parameters

1 Setpoint temperature
2 Ausgangswert m1
3 Temperature tolerance DT
4 Minimum flow mmin
5 Maximum flow mmax
6 Maximum number of iterations maxit

Strings

None

6.2 Block EVAPH

The EVAPH block returns the evaporation enthylpy of steam.

Name	EVAPH
Function	pp0014
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Pressure in Pa

Outputs

1 Evaporation enthalpy of steam in J/kg

Parameters

None

Strings

None

6.3 Block FLDIV

The FLDIV block...

Name	FLDIV
Function	pp0006
Inputs	3
Outputs	4
Parameters	0
Strings	0
Group	S

Inputs

1 Inlet temperature Ti (°C)
2 Inlet masse flow mi
3 Control function γ (determines the distribution of the fluid to the outlets, = fluid fraction for outlet 1)

Outputs

1 Outlet temperature 1
2 Outlet mass flow 1 m1o
3 Outlet temperature 2 T2o
4 Outlet mass flow 2 m2o

Parameters

None

Strings

None

6.4 Block HXCH

The HXCH block...

Name	HXCH
Function	pp0015
Inputs	6
Outputs	6
Parameters	14
Strings	0
Group	S

Inputs

1 Inlet temperature fluid one
2 Mass flow fluid one m1
3 Inlet pressure fluid oneC
4 Inlet temperature fluid two
5 Mass flow fluid two m2
6 Inlet pressure fluid two

Outputs

1 Outlet temperature fluid one
2 Outlet pressure fluid one
3 Outlet enthalpy fluid one
4 Outlet temperature fluid two
5 Outlet pressure fluid two
6 Outlet enthalpy fluid two

Parameters

1 Type of heat exchanger 0: Doubletube counter flow 1: Doubletube parallel flow 2: Rohrbuendel
2 Heat transfer coefficient

3	Heat exchanger area
4	Type of fluid one FL1 0: Wasser 1: Allg. Fluid
5	Temperature 1
6	Specific heat 1 cpf11
7	Temperature 2 TF12
8	Specific heat 2 cpf12
9	Type of fluid two FL2 0: Wasser 1: Allgem. Fluid
10	Temperatur 1 TF21
11	Specific heat 1 cpf21
12	Temperature 2 TF22
13	Specfic heat 2 cpf22
14	Number of iterations

Strings

None

Remarks

Modell im wesentlichen nach Nikolai V. Khartchenko (1995) Thermische Solaranlagen, Berlin / Heidelberg / New York

6.5 Block PCNTRL

The PCNTRL block...

Name	PCNTRL
Function	pp0017
Inputs	3
Outputs	1
Parameters	5
Strings	0
Group	L

Inputs

1 Storage content
2 Former mass flow mp
3 Storage outlet temperature Tout

Outputs

1 Mass flow for power plant

Parameters

1 Intial value for INITOL (OLD!!!!!)
2 mp value
3 Cut in setpoint
4 Cut off setpoint
5 Switch off temperature setpoint

Strings

None

Remarks

6.6 Block PCSSTU

The PCSSTU block...

Name	PCSSTU
Function	pp0018
Inputs	4
Outputs	2
Parameters	5
Strings	0
Group	S

Inputs

1 Temperature T
2 Mass flow m
3 Time t
4 Lower temperature Tmin

Outputs

1 Temperature T
2 Mass flow m

Parameters

1 Specific heat cp
2 Parameter a1 for QStartup
3 Parameter a2 for QStartup
4 Parameter b1 for QStartup (proportional to t)
5 Parameter b2 for QStartup (proportional to t^2)

Strings

None

Remarks

6.7 Block PFSTOR

The PFSTOR block...

Name	PFSTOR
Function	pp0009
Inputs	6
Outputs	4
Parameters	8
Strings	0
Group	S

Inputs

1 Input temperature from collector(-field)
2 Mass flow from collector(-field) mc
3 Input temperature from load TLi
4 Mass flow from load ml
5 Ambient Temperature Ta
6 Zeit t

Outputs

1 Output temperature to collector(-field) Tco
2 Mass flow to collector(-field) mc
3 Output temperature to load TL0
4 Mass flow to load ml

Parameters

1 Volume of tank V
2 Diameter of tank d
3 Specfic heat of oil Cp
4 Density of oil rho
5 Loss coefficient U

6 Temperature of 'empty' tank Te
7 Initial temperature of tank
8 Heat conduction for oil lambda

Strings

None

6.8 Block PIPE

The PIPE block...

Name	PIPE
Function	pp0005
Inputs	5
Outputs	2
Parameters	4
Strings	0
Group	S

Inputs

1 Inlet temperature Ti
2 Fluid mass flow
3 Ambient temperature Ta (°C)

Outputs

1 Outlet temperature
2 Fluid mass flow m

Parameters

1 Pipe diameter d (m)
2 Pipe length l (m)
3 Heat loss coefficient Ud (W/m^2 K)
4 Specific heat cp

Strings

None

6.9 Block RECEIV

The RECEIV block...

Name	RECEIV
Function	pp0003
Inputs	5
Outputs	3
Parameters	9
Strings	0
Group	S

Inputs

1 Concentrated radiation / $W\,m^{-2}$
2 Fluid inlet temperature / °C
3 Fluid mass / $kg\,s^{-1}$
4 Ambient temperature / °C
5 Time / s

Outputs

1 Fluid outlet temperature / °C
2 Fluid mass flow / $kg\,s^{-1}$
3 Chassis temperature / °C

Parameters

1 Area of receiver / m^2
2 Fluid heat capacity / $kJ\,kg^{-1}\,K^{-1}$
3 Convection loss coefficient / $W\,m^{-2}\,K^{-1}$
4 Emissivity of receiver
5 Transmission of cover
6 Absorption of receiver
7 Number of iterations

8 Product of chassis / $J\,K^{-1}$
9 Heat transport coefficient / $W\,K^{-1}\,s^{-1}$

Strings

None

6.10 Block SHADE

The SHADE block...

Name	SHADE
Function	pp0004
Inputs	5
Outputs	1
Parameters	3
Strings	0
Group	S

Inputs

1	Beam radiation on tilted surface Gb
2	Collector azimuth gammac
3	Collector tilt angle β
4	Solar azimuth gammas
5	Solar elevation alphas

Outputs

1	Shaded beam radiation Gbs

Parameters

1	Distance between collectors D
2	Width of collector W
3	Length of collector L

Strings

None

6.11 Block STEAM

The STEAM block...

Name	STEAM
Function	pp0013
Inputs	2
Outputs	4
Parameters	0
Strings	0
Group	S

Inputs

1 Water temperature
2 Water pressure

Outputs

1 Specific entropy
2 Specific enthalpy
3 Specific heat at constant pressure
4 Specific voulme

Parameters

None

Strings

None

6.12 Block STURB

The STURB block...

Name	STURB
Function	pp0016
Inputs	3
Outputs	4
Parameters	4
Strings	0
Group	S

Inputs

1 Mass flow
2 Temperature T
3 Pressure p

Outputs

1 Mechanical power at the turbine shaft Pmech
2 Temperature
3 Pressure p
4 Specific enthalpy h
5 Thermal efficiency eta

Parameters

1 Mass flow at design conditions m0
2 Inlet pressure at design conditions p10
3 Outlet pressure at design conditions p20
4 Polytropic efficiency etap

Strings

None

Remarks

6.13 Block TC2TF

The TC2TF block converts temperatures in degrees Celsius to degrees Fahrenheit.

Name	TC2TF
Function	pp0012
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Temperature in degrees Celsius

Outputs

1 Temperature in degrees Fahrenheit

Parameters

None

Strings

None

6.14 Block TCSTOR

The TCSTOR block...

Name	TCSTOR
Function	pp0008
Inputs	6
Outputs	6
Parameters	8
Strings	0
Group	S

Inputs

1 Input temperature from collector(-field) Tci
2 Mass flow from collector(-field) mc
3 Input temperature from load TLi
4 Mass flow from load ml
5 Ambient temperature Ta
6 Time t

Outputs

1 Output temperature to collector(-field) Tco
2 Mass flow to collector(-field) mc
3 Output temperature to load TL0
4 Mass flow to load ml
5 Speicherinhalt
6 ???

Parameters

1 Tank volume V
2 Number of temperature nodes N
3 Tank diameter d

4 Specfic heat of oil c_p
5 Density of oil ρ
6 Loss coefficient U
7 Number of iterations it
8 Temperature of 'empty' tank Te

Strings

None

6.15 Block TF2TC

The TF2TC block converts temperatures in Fahrenheit to degrees Celsius.

Name	TF2TC
Function	pp0011
Inputs	1
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Temperature in degrees Fahrenheit

Outputs

1 Temperature in degrees Celsius

Parameters

None

Strings

None

6.16 Block TPIECE

The TPIECE block...

Name	TPIECE
Function	pp0007
Inputs	4
Outputs	2
Parameters	0
Strings	0
Group	S

Inputs

1 Inlet temperature 1 (°C)
2 Inlet mass flow 1 m1
3 Inlet temperature 2 T2
4 Inlet mass flow 2 m2

Outputs

1 Outlet temperature (°C)
2 Outlet mass flow mo

Parameters

None

Strings

None

6.17 Block TRACK

The TRACK block...

Name	TRACK
Function	pp0001
Inputs	2
Outputs	3
Parameters	8
Strings	0
Group	S

Inputs

1 Solar azimuth ψ
2 Solar elevation α

Outputs

1 Collector azimuth
2 Collector tilt angle
3 Power of tacking motors

Parameters

1 Tracking mode
  0 Two axis tracking
  1 One axis tracking fixed horizontal axis E-W
  2 One axis tracking fixed horizontal axis N-S
2 Minimum elevation
3 Minimum azimuth
4 Maximum azimuth
5 Tracking motor power hozizontal
6 Tracking motor power vertical
7 Park position horizontal

8	Park position vertical

Strings

None

6.18 Block TROUGH

The TROUGH block...

Name	TROUGH
Function	pp0002
Inputs	6
Outputs	2
Parameters	6
Strings	0
Group	S

Inputs

1 Beam radiation onto aperture Gb (W/m^2)
2 Diffuse radiation onto aperture Gd (W/m^2)
3 Collector azimuth γ (°)
4 Collector tilt angle β (°)
5 Solar azimuth ψ (°)
6 Solar elevation α (°)

Outputs

1 Concentrated radiation (W/m^2)
2 Intercept factor

Parameters

1 Area of aperture / m^2
2 Width of aperture / m
3 Area of receiverm / m^2
4 Width of receiver / m
5 Reflectivity of mirrors
6 Rim angle / degrees

Strings

None

7 :: Community blocks

7.1 Block 2pcon2

An on–off controller (2 step or bang-bang). Experimental block. Testing needs to be done.

2PCON2

Name	2pcon2
Function	onoffcontroller
Inputs	2
Outputs	1
Parameters	3
Strings	0
Group	3

Inputs

1 Control temperature [Celsius]
2 Mass flow on [kg/s]

Outputs

1 Mass flow [kg/s]

Parameters

1 Set-point temperature [Celsius]
2 deltaT [K]
3 Mass flow off [kg/s]

Strings

None

7.2 Block UBBHKWSG

Calculates the performance of a CHP system

This is the first version of UBBHKWSG by Maryam Zirak in January 2021. SG (BHKWWG) stands for *Stromgeführt* (Electricity driven). In this version, thermal and electrical efficiency are defined as parameter.

The efficiency curve in partial operation is defined as quadratic equation, $ax^2 + bx + c$. The coefficients, a,b and c, are defined as parameters in this model.

For heat-driven CHP, see UBBHKWWG.

UBBHKWSG

Name	UBBHKWSG
Function	ubbhkwsg
Inputs	3
Outputs	5
Parameters	14
Strings	0
Group	S

Inputs

1 Power Demand, kW
2 inlet Water temperature, feed to CHP, degree C
3 Output Water temperature,targeted for the heating network, degree C

Outputs

1 Mass flow of water,kg per hour
2 Heat generation, kW
3 Electric generation, kW
4 Fuel demand, cubic meters
5 Water temperature, outlet, degree C

Parameters

1	Nominal electric output performance, kW
2	Nominal heat output, kW
3	lower limit for output thermal power, per cent of nominal maximum thermal output
4	Coefficient a_1 for electrical efficiency (ele) curve when ele=$a_1x^2 + b_1x + c_1$ and x is the ratio of electrical output to nomminal electrical output
5	Coefficient b_1 for electrical efficiency (ele) curve when $ele = a_1x^2 + b_1x + c_1$ and x is the ratio of electrical output to nomminal electrical output
6	Coefficient c_1 for electrical efficiency (ele) curve when $ele = a_1x^2 + b_1x + c_1$ and x is the ratio of electrical output to nomminal electrical output
7	Coefficient a_2 for thermal efficiency (n_{th}) curve when $n_{th} = a_2x^2 + b_2x + c_2$ and x is the ratio of electrical output to nomminal electrical output
8	Coefficient b_2 for thermal efficiency (n_{th}) curve when $n_{th} = a_2x^2 + b_2x + c_2$ and x is the ratio of electrical output to nomminal electrical output
9	Coefficient c_2 for thermal efficiency (nth) curve when $n_{th} = a_2x^2 + b_2x + c_2$ and x is the ratio of thermal output
10	thermal efficiency at full load operation, per cent
11	electrical efficiency at full load operation, per cent
12	Heat capayity of Water or medium in CHP, kJ per kg per K
13	Fuel lower heating value, kWh per kg
14	Gas density, kg per cubic meters

Strings

None

7.3 Block UBBHKWWG

Calculates the performance of a CHP system

This is a modifed version of UBBHKWNG by Maryam Zirak in Oktober 2020. WG (BHKWWG) stands for *Wärmegeführt* (heat-driven). In this modification, thermal and electrical efficiency are defined as parameter.

The efficiency curve in partial operation is defined as quadratic equation, $ax^2 + bx + c$. The coefficients, a,b and c, are defined as parameters in this model.

For electricity-driven CHP, see UBBHKWSG.

UBBHKWWG

Name	UBBHKWWG
Function	ubbhkwwg
Inputs	3
Outputs	5
Parameters	13
Strings	0
Group	S

Inputs

1 Heat Demand, kW
2 inlet Water temperature, feed to CHP, degree C
3 Output Water temperature,targeted for the heating network, degree C

Outputs

1 Mass flow of water,kg per hour
2 Heat output, kW
3 Electric performance, kW
4 Fuel demand, cubic meters
5 Water temperature, outlet, degree C

Parameters

1	Nominal electric output performance, kW
2	Nominal heat output, kW
3	lower limit for output thermal power, per cent of nominal maximum thermal output
4	Coefficient a_1 for electrical efficiency (ele) curve when ele=$a_1x^2 + b_1x + c_1$ and x is the ratio of electrical output to nomminal electrical output
5	Coefficient b_1 for electrical efficiency (ele) curve when $ele = a_1x^2 + b_1x + c_1$ and x is the ratio of electrical output to nomminal electrical output
6	Coefficient c_1 for electrical efficiency (ele) curve when $ele = a_1x^2 + b_1x + c_1$ and x is the ratio of electrical output to nomminal electrical output
7	Coefficient a_2 for thermal efficiency (n_{th}) curve when $n_{th} = a_2x^2 + b_2x + c_2$ and x is the ratio of electrical output to nomminal electrical output
8	Coefficient b_2 for thermal efficiency (n_{th}) curve when $n_{th} = a_2x^2 + b_2x + c_2$ and x is the ratio of electrical output to nomminal electrical output
9	Coefficient c_2 for thermal efficiency (nth) curve when $n_{th} = a_2x^2 + b_2x + c_2$ and x is the ratio of thermal output
10	thermal efficiency at full load operation, per cent
11	electrical efficiency at full load operation, per cent
12	Fuel lower heating value, kWh per kg
13	Gas density, kg per cubic meters

Strings

None

7.4 Block UBH2COMPRESSOR

Name	UBH2COMPRESSOR
Function	ubh2compressor
Inputs	5
Outputs	2
Parameters	5
Strings	0
Group	S

Inputs

1 Inflow [kg/timestep]
2 Inflow Pressure [MPa]
3 Ambient Temperature [degree C]
4 Pressure Setpoint [MPa]
5 Storage Pressure [MPa]

Outputs

1 Outflow [kg/timestep]
2 Compressor Power [W]

Parameters

1 Number of Stages
2 Thermodynamic Mode
3 Max. Pressure [MPa]
4 Max. Delivery Rate [kg/h]
5 Time Step

Strings

None

7.5 Block UBH2CYLINDER

Name	UBH2CYLINDER
Function	ubh2cylinder
Inputs	4
Outputs	4
Parameters	7
Strings	0
Group	S

Inputs

1 Inflow [kg/tinestep]
2 Time
3 Ambient Temperature [°C]
4 Outflow Setpoint [kg/tinestep]

Outputs

1 Filling Level [MPa]
2 Stored Hydrogen [kg]
3 Outflow [kg/timestep]
4 Number of Refills [-]

Parameters

1 Amount of Cylinders
2 Cylinder Volume [m^3]
3 Time Step
4 Initial Hydrogen Pressure [MPa]
5 Equation of State
6 Automatic Refills
7 Automatic Refills Lower Boundary [MPa]

Strings

None

7.6 Block UBISONLAND

This block rounds the coordinates to the nearest °, and checks with an internal table if the corresponding point is probably on land or not.

This same function is used by the meteorology blocks (e.g. MTMLALO2) to check if the correct convention is used (48.77 -9.18 is in the ocean, 48.77 9.18 is on land)

UBISONLAND

Name	UBISONLAND
Function	ubisonland
Inputs	2
Outputs	1
Parameters	0
Strings	0
Group	S

Inputs

1 Latitude
2 Longitude (east-positive)

Outputs

1 Are the coordinates on land?

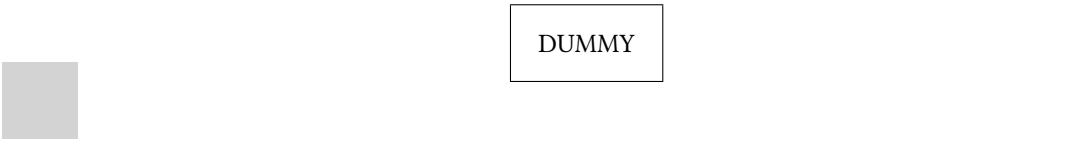
Parameters

None

Strings

None

7.7 Block UBPEMECSTACK



Name	UBPEMECSTACK
Function	ubpemecstack
Inputs	4
Outputs	4
Parameters	11
Strings	0
Group	S

Inputs

1 Current [A]
2 Time
3 Ambient Temperature / Temperature Setpoint [Celsius]
4 Heating Power [W]

Outputs

1 Stack Voltage [V]
2 Cell Temperature [°C]
3 Hydrogen Mass Flow [kg/timestep]
4 Heat Generation [W]

Parameters

1 Cell Area [cm^2]
2 Amount of Cells per Stack
3 Heat Capacity [J/K]
4 Heat Transfer Coefficient
5 Heat Transfer Exponent
6 Time Step
7 Limitting Current Density
8 Anode Charge Transfer
9 Cathode Charge Transfer

10 Anode Exchange Current Density
11 Cathode Exchange Current Density
12 Thermal Model

Strings

None

7.8 Block UBPEMFC

Name	UBPEMFC
Function	ubpemfc
Inputs	4
Outputs	6
Parameters	11
Strings	0
Group	S

Inputs

1 Current [A]
2 Time
3 Ambient Temperature / Temperature Setpoint [Celsius]
4 Heating Power [W]

Outputs

1 Stack Voltage [V]
2 Stack Power [W]
3 Hydrogen Consumption Rate [kg/timestep]
4 Air Consumption Rate [kg/timestep]
5 Cell Temperature [°C]
6 Heat Generation [W]

Parameters

1 Cell Area [cm^2]
2 Number of Cells
3 Hydrogen Pressure [atm]
4 Air Pressure [atm]
5 Ohmic Resistance
6 Exchange Current Density
7 Limitting Current Density

8	Time Step
9	Heat Capacity [J/K]
10	Heat Transfer Coefficient
11	Heat Transfer Exponent
12	Thermal Model

Strings

None

7.9 Block UBSTORAGE

UBSTORAGE

Name	UBSTORAGE
Function	ubstorage
Inputs	2
Outputs	7
Parameters	9
Strings	0
Group	S

Inputs

1 Energy surplus
2 Energy deficiency

Outputs

1 State of charge
2 To battery
3 From battery
4 Capacity
5 Number of cycles
6 Energy losses
7 Energy losses
8 Capacity of reference

Parameters

1 Battery capacity
2 Initial capacity
3 Charging efficiency
4 Discharging efficiency
5 Time step
6 DOD
7 Self discharge rate per month

8 Inverter efficiency
9 Ageing coefficient

Strings

None

Index

2pcon2, 779

ABS, 5
ACC, 6
ACCC, 8
ACCP, 10
ACM, 545
ACOS, 12
ACOT, 14
AIR, 548
AIRHEAT, 550
AM, 299
AMBIENT, 685
AND, 16
ANINT, 18
ASIN, 20
ATAN, 22
ATAN2, 24
ATEND, 25
ATM1976, 301
ATT, 27
AVE, 28
AVEC, 30
AVEM, 32
AVEP, 34
AVEW, 36

BALON, 429
BASE, 687
BCR, 432
BDC, 435
BETAMAX, 303
BETAOPT, 305
BETGAM, 307
BTI, 437
BTV, 441
BUCKET, 444

CABLE, 446
CALEND, 689
CHARTXY, 37
CHS, 38
CLOCK, 39
CLOCKL, 42
CNTRL, 753
COMPRS, 448
CONST, 44
CONTRL, 450
CONZON, 690
COOLTW, 555
COS, 45
COSH, 47
COT, 49
COTH, 51
COWV, 309
CPUMP, 452
CSCH, 53
CTRHYS, 560
CTRL4DY, 562
CTRLDA, 564
CTRLROOM, 692
CUM, 55
CUMC, 57
CUMP, 59
CUTOFF, 454
CYCLEF, 61

D18599, 694
DAYLC, 698
DECCTL, 566
DECCTY, 569
DEG2RAD, 63
DELAY, 64
DELAYS, 66
DHYD, 571
DIFF, 68
DINCLI, 311
DIODEI, 456
DIODEV, 458
DISEXP, 69
DISGAU, 71
DISGR, 312
DISRAY, 73
DISWEI, 75
DIV, 77
DLC, 79
DO, 81
DOW, 83

DOY, 85
DPBOWR, 573
DPBRAR, 575
DPCH, 577
DPDIF, 579
DPIN, 581
DPJUNR, 583
DPM, 87
DPOUT, 585
DPPARA, 587
DPRED, 589
DPTUBE, 591
DPZETA, 593
DSTAR, 595
DWHEEL, 597
DYBASE, 700
DYNDGL, 703

E, 89
E0, 314
ECV, 460
EINZON, 705
ENERBA, 708
EQ, 90
ERF, 92
ESCOOL, 599
ESP, 466
ET, 316
ETAIS, 468
EVAPH, 754
EVCOOL, 601
EXP, 94
EXPG, 96
EXPRESSION, 98

F, 99
F90, 710
F90P, 711
FACADE, 713
FADYBS, 714
FCV, 469
FDBS, 717
FDIST, 100
FERMID, 474
FFT, 102
FIT3PC, 720
FIT3PH, 723
FIT4PC, 724
FIT4PH, 725
FIT5PH, 726
FITEXP, 104
FITLIN, 106
FITLLS, 109
FITLN, 110
FITPOW, 112
FITWEI, 114
FLDIV, 755
FPARA, 727
FPARAP, 728
FRAC, 116

G2GDD, 319
G2GDH, 322
G2GDM, 325
GAIN, 118
GASDEV, 119
GBN2GBT, 328
GE, 121
GENG, 330
GENG2, 333
GENGD, 335
GENGD2, 338
GENGH, 340
GENGH2, 343
GENGT, 345
GENGT2, 348
GENV, 350
GH2GT, 352
GH2GT2, 355
GMEAN, 357
GMIX, 729
GOH, 360
GOH2, 364
GON, 365
GR, 603
GREGOR, 123

GS, 125
GT, 126
GTLOSS, 367

H, 128
H2OHE, 605
H2OHL, 607
H2OHV, 609
HBAR, 129
HOY, 130
HPPOLY, 476
HR, 731
HUMA2R, 611
HUMH, 613
HUMR2A, 615
HXC, 617
HXCH, 756
HXEARTH, 619
HXS, 622
HXXI, 624
HYSTA, 132
HYSTC, 134

IF, 136
IFELSENAN, 138
IFNEG, 139
IFPOS, 140
INANGE, 370
INANGH, 372
INFINITY, 141
INT, 142
INTGRL, 144
INV, 146
IRADIA, 732
IVETAEU, 478
IVP, 480
IVPFIT, 482
IVPV, 484

JULIAN, 148

K, 150
KN, 374

LBWH, 627
LBWP, 629
LBWT, 631
LBWX, 633
LE, 151
LN, 153
LOG, 155
LOOP, 157
LT, 160

MAE, 162
MAM, 486
MAX, 163
MAXX, 165
MAXXC, 166
MAXXP, 168
MAXXXY, 170
MAXXXYC, 172
MAXXXYP, 174
MBE, 176
MBOX, 177
MBROT, 179
MGP, 488
MIN, 181
MINN, 183
MINNC, 185
MINNP, 187
MINNXY, 189
MINNXYC, 191
MINNXYP, 193
MIXER, 635
MIXQHC, 637
MOD, 195
modified one diode model, 496
MOONAE2, 376
MOY, 196
MPLEX, 198
MPP, 200
MTM, 378
MTM2, 381
MTMLALO, 382
MTMLALO2, 384
MTMUP, 386

MUL, 203

NAN, 204
NE, 205
NIEG, 490
NOP, 207
NOW, 208
NULL, 210

OFFSET, 212
OR, 213

PCNTRL, 758
PCSSTU, 759
PFSTOR, 760
PHIOPT, 389
PI, 215
PID, 216
PIPE, 762
PLANCK, 391
PLOT, 218
PLOTP, 222
PLOTPM3D, 224
PLOTPMC, 226
POLYG, 228
POLYG2, 230
POLYGS, 233
POLYN, 235
PVA2C, 492
PVAI, 494
PVAV, 498
PVDET1, 501
PVDET2, 506
PVECON, 510
PVFIT1, 513
PVFIT2, 517
PVFITA, 521
PVI, 526
PVT, 639
PVTCOOL, 642
PVV, 532

Q, 237
Q3PC, 733
Q3PH, 734
Q4PC, 735
Q4PH, 736
Q5P, 737
QDTV, 738

R134A, 645
RAD2DEG, 238
RADI, 739
RAN1, 239
READ, 241
READD, 243
READN, 246
RECEIV, 763
REPORT, 248
RMAE, 251
RMBE, 252
RMSE, 253
ROOM, 741
ROOT, 254
RRMSE, 256

SATP, 646
SATT, 648
SCAIRC, 650
SCAIRCD, 653
SCDYN, 657
SCETA, 660
SCH, 257
SCHEDU, 743
SCREEN, 259
SCREEN1G, 261
SEDES3, 393
SHADE, 765
SHADF, 395
SIGMA, 262
SIN, 263
SINH, 265
SKYALL, 397
SKYC, 398
SKYIC, 400
SKYIM, 402
SKYIO, 404

SKYO, 406
SOY, 267
SPEAR, 269
SQRT, 270
STDEV, 272
STDEVC, 275
STDEVP, 277
STEAM, 766
STOP, 279
STORE, 535
STURB, 767
SUM, 281
SUMP, 282
SUNAE, 408
SUNAE2, 411
SUNAEA, 413
SUNAEA2, 414

TAN, 284
TANH, 286
TANKFM, 663
TANKST, 665
TAUW, 415
TC2TF, 768
TCSTOR, 769
TDEW, 417
TF2TC, 771
THREEV, 668
TMEAN, 419
TOL, 288
TPIECE, 772
TRACK, 773
TROUGH, 775
TSKY, 421
TSOIL, 423
TUBLO, 670
two diode model, 528

UBBHKWSG, 780
UBBHKWWG, 782
UBH2COMPRESSOR, 784
UBH2CYLINDER, 785
UBISONLAND, 787
UBPEMECSTACK, 788
UBPEMFC, 790
UBSTORAGE, 792
ULC, 289

VBI, 538
VBOWR, 671
VDIF, 673
VECTOR, 291
VIN, 675
VOUT, 677
VRED, 679
VZETA, 681

WALL, 744
WALLX, 746
WINDOW, 748
WRITE, 292

XOR, 295
XSNOFX, 540